

Projet d'intelligence artificielle, partie 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1																									
2																									
3																									
4	O																								
5																									
6																									
7				x																					
8										O										O				O	
9																									
10																									
11														O											
12																									
13																									
14																								O	
15													O												
16																									
17																									
18																									
19												O													
20										O										O					
21																									
22		O																							
23																									
24										O										O					
25																			O						

Figure 1 : schéma de l'entrepôt, en bleu les rayonnages, en vert la zone de livraison. Le x est un objet, les O sont des chariots.

Une grande entreprise de commerce dispose d'un grand entrepôt pour stocker les objets qu'elle vend. Pour gagner en place, les objets sont rangés eu fur et à mesure là où la place existe, de manière non ordonnée. Une gestion numérique rigoureuse permet cependant de mémoriser la localisation exacte de chaque objet. Le schéma de l'entrepôt est présenté ci-dessus. Lorsqu'un objet est vendu, il suffit donc d'avoir sa référence et de questionner la base de données pour connaître l'emplacement où il faut le récupérer, afin de préparer la livraison. Cet emplacement est défini par des coordonnées (x,y) dans l'entrepôt, une orientation k (nord ou sud) et une hauteur z. Par exemple, pour aller chercher le X, il faut aller en (4,6), s'orienter au sud et faire monter le chariot à une hauteur donnée. L'entreprise souhaite automatiser la récupération. Elle envisage l'utilisation de chariots intelligents, capables de se déplacer horizontalement ou verticalement dans l'entrepôt en suivant de manière précise des lignes au sol et récupérer l'objet à l'aide d'un bras manipulateur. Lorsque le chariot doit tourner, il effectue une rotation sur lui-même et n'empiète pas sur les lignes où pourrait se trouver un autre chariot. Il n'y a jamais de déplacement en diagonal. La contrainte la plus importante est qu'il ne peut y avoir 2 chariots au même endroit (mêmes coordonnées), ce qui implique que quand un chariot s'engage dans une allée, il bloque l'accès.

L'objectif est de réaliser un programme qui va gérer la récupération des objets dans l'entrepôt.

Question préliminaire :

Initialisez votre programme avec les données du problème :

- créez un tableau d'entiers pour représenter l'entrepôt. Les cases correspondant aux rayons seront marquées -1 (obstacle). Affichez le tableau à l'écran.

- l'utilisateur doit indiquer le nombre de chariots utilisés. Appelons ce nombre NBC. Créez une liste de NBC chariots et demandez à l'utilisateur de définir leur position initiale dans l'entrepôt. Prévoyez des valeurs par défaut pour gagner du temps lors des tests ...

Question 1 : Demandez à l'utilisateur de choisir un chariot en particulier et d'indiquer la position (x,y,k,z) d'un objet à aller chercher.

- a) Exploitez les fichiers sources qui vous sont donnés (GenericNode.cs et graph. cs) pour déterminer le plus court chemin que doit suivre le chariot pour récupérer l'objet en évitant les obstacles constitués par les rayons et les autres chariots (on les suppose statiques pour le moment). Affichez le plus court chemin à l'écran.
- b) Proposez une heuristique pour réduire les temps de calculs et implémentez-la.

Question 2 : Lorsqu'un chemin est déterminé, on lui associe une trajectoire spatiotemporelle, caractérisée de la manière suivante. A chaque case (x,y) , on associe une date de présence. Lorsqu'il se déplace en gardant la même direction, il met précisément une seconde pour passer à la case suivante. En revanche, s'il est amené à changer de direction, il reste au même endroit pendant 3 secondes, le temps de se réorienter et ensuite seulement il change de case à la seconde suivante. Lorsqu'il arrive en (x,y) , le chariot met 2 secondes multipliées par la hauteur h (un nombre entre 0 et 10) pour atteindre la position h demandée. Ensuite, il lui faut encore 10 secondes pour récupérer l'objet, quelle que soit sa situation (nord ou sud). Enfin, le chariot redescend et suit une autre trajectoire pour amener son colis en zone de livraison (colonne 1), en respectant les mêmes contraintes temporelles. Déterminez le plus court chemin, non pas relativement à la distance, mais relativement au temps pour que le chariot aille chercher un objet donné et le ramène sur une case quelconque de la colonne 1. On supposera pour le moment que les autres chariots ne bougent pas.

Question 3 : Les autres chariots ne sont en général pas statiques. Pour déterminer si une case sera accessible à la date demandée, il faut connaître la trajectoire spatiotemporelle des autres chariots. Pour cela, on va mémoriser dans le programme la liste des chariots et la trajectoire qu'ils ont planifiée. Il n'y a 2 cas possibles :

- soit il est sur la colonne 1 à l'arrêt (pas d'objet à aller chercher, ou il est arrivé et en train de livrer son colis). Dans ce cas, il est supposé rester à cette position.
- soit il a planifié une trajectoire et dans ce cas, il est possible de savoir à quel endroit il sera à tout moment dans le futur.

Ayant accès à ces informations, programmez un chariot pour qu'il détermine le plus court chemin temporel en tenant compte de ces contraintes. Notez qu'un chariot peut faire du surplace, à condition bien sûr que cette même case envisagée une seconde plus tard soit toujours inoccupée.

- a) Pour tester votre programme, vous pouvez placer tous les chariots sur la colonne 1 pour commencer, puis leur attribuer au fur et à mesure un objet à aller chercher. La cohérence du système est garantie par le fait qu'une nouvelle attribution tient compte de toutes les attributions déjà effectuées.
- b) Essayez d'optimiser votre code en ajoutant des heuristiques. Par exemple, si un chariot est déjà présent dans une allée et bloque le passage, il est évident que le temps restant est bien plus important que s'il n'y était pas ...