

Gestion d'un entrepôt

Objectif

Optimiser le parcours des transpalettes dans un entrepôt afin de maximiser le remplissage de camions de transports partant d'un entrepôt.

Présentation

En entrée du programme un fichier texte sera fourni décrivant les différentes contraintes concernant la modélisation des différentes données du problème à optimiser.

L'entrepôt peut être vu comme une grille contenant les informations suivantes :

- position des colis à livrer dans le camion ainsi que leur poids représenté par une couleur (**GREEN**, **YELLOW** ou **BLUE**)
- position du ou des transpalettes disponibles
- position du camion à remplir
- poids maximale transportable par le camion (charge maximale)
- nombre de tour avant qu'un camion soit disponible
- nombre de tours total à simuler

Voici un exemple visuel de représentation du problème :

transpalette 1				
		colis 1		colis 4
		colis 2		
colis 3				



Description du fichier en entrée

La première ligne contiendra 3 nombre séparés par un espace : nombre de cases en largeur de l'entrepôt, nombre de cases en longueur de l'entrepôt, nombre de tours à simuler.

Les n lignes suivantes décriront les colis présents dans l'entrepôt. Chaque ligne contiendra une chaîne de caractères (sans espace) représentant le nom du colis suivie de sa position (2 nombres), suivie de sa couleur (en majuscule ou en minuscule).

Les k lignes suivantes décriront les transpalettes présents dans l'entrepôt. Chaque ligne contiendra une chaîne de caractères (sans espace) représentant le nom du transpalette, suivie de sa position (2 nombres).

Enfin la dernière ligne du fichier sera composée de 4 nombres. Les 2 premiers nombres représenteront la position de l'entrée du camion, le 3ème représentera sa charge maximale, et le dernier représentera le nombre de tours avant qu'un nouveau camion soit disponible.

Voici un exemple de fichier avec 4 colis, 1 transpalette et un entrepôt de forme carré :

```
5 5 1000
colis_a_livrer 2 1 green
paquet 2 2 BLUE
deadpool 0 3 yellow
colère_DU_dragon 4 1 green
transpalette_1 0 0
3 4 4000 5
```

Définition des contraintes métier

Une position est définie par une coordonnée. Une coordonnée est une paire de nombres entier et positif (0-n). Le premier nombre représente les abscisses et le second les ordonnées. La position [0,0] est située en haut à gauche de l'entrepôt.

Un colis est défini par une position et une couleur. Il y a trois couleurs différentes (**YELLOW**, **GREEN** et **BLUE**) représentant chacune un poids différent.

Le **YELLOW** vaut 100 kg, le **GREEN** 200 kg et le **BLUE** 500 kg.

Un transpalette ne peut porter qu'un colis à la fois et n'a pas de contraintes de poids. Il peut donc porter les 3 types de colis. Il est défini par une position et peut donc contenir un colis ou pas. Il ne peut se déplacer que d'une case à chaque tour et ne peut pas aller en diagonale. Une fois devant le camion il peut décharger sa livraison dans celui-ci (cette action coûtera un tour). Pour prendre un colis, un transpalette devra se trouver à côté d'un colis (une des 4 directions adjacentes, hors diagonale). Il faudra un tour pour prendre le colis de son emplacement.

Un camion sera défini par une position et une valeur entière strictement positive représentant sa capacité maximale (par exemple 1000 kg). La position représentera la case où le transpalette devra être présent pour vider sa marchandise (si on prend l'exemple précédent la position du camion sera [3,4]).

Un camion pourra soit attendre ("**WAITING**"), soit partir ("**GONE**"). Le camion restera dans l'état "**GONE**" jusqu'à l'arrivée d'un autre camion. Ce paramètre correspond au paramètre "nombre de tours avant qu'un camion soit disponible". Il n'y aura qu'un camion au maximum devant l'entrepôt et il attendra toujours au même endroit.

Un transpalette ne pourra occuper une case où se trouve déjà un colis ou un autre transpalette.

Le nombre de tours est un entier positif compris entre 10 et 100 000.

Définition des contraintes applicatives

Le programme se terminera quand le nombre total de tours sera atteint ou quand il n'y aura plus de colis à livrer (le dernier colis est livré dans un camion).

A chaque tour le programme affichera

- le tour en cours
- pour chaque transpalette l'action choisie et la nouvelle position si celui-ci se déplace
- pour le camion, l'action choisie

Les actions possibles pour un transpalette sont :

- **GO**: cette action signifie que celui-ci se déplace, il sera suivi de sa nouvelle coordonnée
- **WAIT**: le transpalette ne bouge pas.
- **TAKE**: le transpalette prend un colis. Le colis pris sera affiché (nom + couleur).
- **LEAVE**: le transpalette dépose le colis dans le camion. Le colis pris sera affiché (nom + couleur).

Les états possible pour un camion :

- **WAITING**: le camion attend d'être chargé. On affichera sur la même ligne son poids maximal ainsi que la somme des colis déjà chargés.
- **GONE**: le camion est parti. On affichera sur la même ligne son poids maximal ainsi que la somme des colis déjà chargés.

Un nouveau saut de ligne sera effectué entre chaque tour.

A la fin du programme un emoji sera affiché représentant la condition d'arrêt:

- 🤖 : l'entrepôt est vide.
- 😊 : le nombre de tours max est atteint et il reste des colis à livrer.
- 🤯 : suivi d'un message d'erreur dans le cas d'un problème de données non valides ou de crash du programme.

Exemple de sorti console :

```
tour 1
transpalette_1 GO [1,0]
transpalette_2 GO [2,2]
camion WAITING 1000 0

tour 2
```

```
transpalette_1 TAKE colis_1 BLUE
transpalette_2 WAIT
camion WAITING 1000 0

...

tour 29
transpalette_1 WAIT
transpalette_2 LEAVE colis_2 YELLOW
camion WAITING 1000 800

tour 30
transpalette_1 WAIT
transpalette_2 WAIT
camion GONE 1000 900
```



Livrable attendu

Les sources du projet devront être disponibles sur Github ou Gitlab. Le repository devra contenir un fichier README.md ayant une section expliquant comment compiler et lancer le projet. Une autre section décrira l'organisation des sources et des packages. Enfin une dernière section décrira la stratégie retenue.

Bonus

Ce projet va vous permettre de jouer avec le langage GO, vous pouvez donc agrémenter ce projet d'une interface graphique (native à l'OS), de scripts de déploiement sur le cloud (GCP, AWS, AZURE, ...), de génération d'un serveur web avec interface graphique (HTML + Javascript), ...

Happy Coding 🎉