
Travail : Bloc A - Fondements

8INF957 : Programmation objet avancée

Pondération : 10%

Consignes :

- Travail en équipe de 1 ou 2
 - Date de remise : semaine 6 (6 octobre)
 - Langage : Java
 - Remise par Teams
 - Un seul projet, compilé et fonctionnel
 - La diagramme de classes de votre solution
-



eCommerce¹

En 2500, le commerce dans l'espace est essentiel entre les planètes de la Fédération. Les vaisseaux voyagent d'une planète à l'autre dans un cycle sans fin. À chaque port, on charge des produits variés et on les décharge dans le port suivant.

Un vaisseau spatial

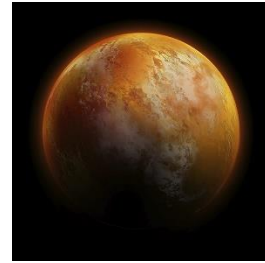
- Il en existe plusieurs types (pour simplifier notre exemple, disons 3)
- Chaque vaisseau peut embarquer certains produits déterminés lors sa création et a une certaine capacité de charge (Par exemple, le vaisseauA peut transporter 30T de produit1, 20T de produit2 et 10T de produit3). Il ne peut charger que ces produits
- Il existe deux grandes catégories de produits (dangereux et régulier)
- Chaque vaisseau spatial a son itinéraire particulier qui est fixé dynamiquement lors de sa création et ne change pas
- Un vaisseau peut charger des produits et décharger des produits.
- On peut lui demander de fournir un « bilan des transports » qui affiche toutes les transactions effectuées.
- Un vaisseau fait du commerce un certain nombre de cycles. Par la suite, il quitte le eCommerce (mais sera toujours inclus dans le rapport bilan)
- Vous pouvez apporter les ajouts au besoin



¹ eCommerce : espace commerce

Une planète

- Chaque planète possède des ressources différentes (des produits) en quantité/limite variables définies lors du chargement de « l'espace commercial »
- Une planète vend des produits qui sont chargés dans les vaisseaux
- Une planète achète des produits qui sont déchargés des vaisseaux
- On peut lui demander de fournir un « bilan des transactions » qui affiche toutes les transactions effectuées.
- Les planètes ne sont pas liées entre elles
- Chaque planète possède entre un et trois ports et ne peut accueillir qu'un vaisseau par port (*et un en attente. Voir plus loin*)
- Vous pouvez apporter des ajouts au besoin



L'espace commercial

- Il existe plusieurs planètes (au moins 5 dans notre exemple)
- À chaque cycle, de nouveaux vaisseaux peuvent apparaître pour faire du eCommerce
- Le cycle est l'unité de mesure (un tour de boucle dans votre projet?)
- Voici une idée de l'algorithmique du projet (**suggestion seulement**)

```
Chargement et initialisation des planètes et des vaisseaux
Lecture du nombre de cycles
POUR chaque cycle
    Afficher les bilans (optionnel : pour vous aider)
    Arrivée/départ des vaisseaux
    Déplacer les vaisseaux
    Décharger les vaisseaux
    Charger les vaisseaux
    (Mettre une pause ici au besoin)
Afficher les bilans
```

Voici quelques règles à respecter

- Règle #1 : Un vaisseau/une planète ne peut pas charger plus de produits que sa capacité
- Règle #2 : On tente toujours de chercher à avoir la plus grande quantité de produits chargée, déchargée lors d'une transaction
- Règle #3 : On complète les transactions avec un vaisseau avant de passer au suivant
- Règle #4 : On complète les transactions sur une planète avant de passer à la suivante
- Règle #5 : Si la capacité maximum des ports d'une planète est atteinte, les vaisseaux doivent demeurer « en attente » d'une place et pourront accéder au port dans un cycle suivant

- Règle #6 : Un port peut accueillir un seul vaisseau et un second peut être placé en attente. Si la capacité d'accueil d'une planète est dépassée, le vaisseau doit quitter abruptement l'espace commercial (géré ce cas comme une **exception commerciale**)
- La durée des opérations (il n'y a pas de cycle partiel)
 - Les produits dangereux
 - Charger 20T (1 cycle)
 - Décharger 10T (1 cycle)
 - Les produits régulier
 - Charger 50T (1 cycle)
 - Décharger 25T (1 cycle)
 - Déplacement entre les planètes (1 cycle)

Les contraintes

- Utilisation des objets (*concept*)
- Appliquer le modèle GRASP (chaque classe a sa propre responsabilité)
- Intégrer une gestion des accidents commerciaux
- gestion tour par tour, pas d'interface graphique, seulement en console mais vous pouvez utiliser un fichier si c'est plus simple pour vous

Dernières notes

- *L'objectif de ce laboratoire est de valider votre capacité à mettre en pratique les notions d'abstraction, de découpage de vos objets, de gestion des exceptions personnalisées mais surtout votre compétence à créer un modèle sécuritaire, fiable et facile à comprendre*
- *Fournir le diagramme de classes (png/jpg) ainsi qu'un projet fonctionnel*
- *N'hésitez pas à ajouter des preuves de fonctionnement de votre projet (capture d'écran ou fichier), juste en cas ou je ne serais pas en mesure de l'exécuter*