8INF259 : STRUCTURES DE DONNÉES

TP3 : Utilisation des graphes

Travail seul ou en équipe de 2

À remettre au plus tard le 28 avril 2016 à 23h59

1.Objectifs

Ce devoir a pour but :

- De se familiariser avec l'utilisation des graphes ;
- De consolider les connaissances acquises lors des cours théoriques ;
- De se familiariser avec quelques algorithmes vus en classe.

2. Mise en contexte

L'annonce du prochain jeu vidéo de la populaire série « *Vitesse spatiale »* fait sensation dans le monde des jeux vidéos. Il s'agit d'un jeu de simulation où le joueur doit faire la gestion du transport spatial entre les planètes d'un lointain système stellaire. Le joueur doit ainsi minimiser les coûts de sa flotte de vaisseau spatial en optimisant les routes de transport et ainsi gagner plus d'argent pour s'acheter de meilleur vaisseau… Mais attention, les conflits entre les différentes nations viendront compliquer les choses dans ce jeu de simulation *spatiopolitique* ! Le principe du jeu est simple :

- Chaque type de vaisseau possède une capacité de carburant fixe. Il est prévu d'avoir différents types de vaisseau avec des apparences plus intéressantes les unes que les autres et des capacités de carburant variées.
- Chaque planète a un nom, des coordonnées (X, Y), une nation, une population ainsi qu'un prix de carburant.
- L'objectif d'un vaisseau est de se rendre d'une planète à une autre. Pour arriver à destination, il doit planifier son trajet en prévoyant les planètes où il devra faire le plein. Il doit être en mesure d'optimiser ses coûts, mais aussi la distance parcourue.
- À chaque planète, un vaisseau prend seulement le carburant qu'il a besoin pour se rendre à la prochaine planète de son parcours.
- Des scénarios de guerre entre différentes nations peuvent affecter les routes commerciales déjà établies.
- Un mécanisme de gestion des ressources rocambolesques et intrigants qui fait de ce jeu le jeu le plus fantastique du vingt et unième siècle. *Cette fonctionnalité est non incluse dans le devoir*

3. Travail à effectuer

On vous demande de participer à la programmation du moteur du jeu « *Vitesse spatiale ».* Vous devez ainsi mettre en place la structure de données qui permettra au joueur d'évoluer dans l'environnement spatial.
Votre programme doit pouvoir :

- Charger le fichier des planètes dans une structure de données appropriées ;
- Charger le fichier de définitions des types de vaisseau ;
- Sélectionner un type de vaisseau, une planète source, une planète destination et définir :
    - **S'il existe une route entre les planètes pour ce vaisseau**, sachant que la distance entre deux planètes est limitée par la capacité de carburant du vaisseau ;
    - **Le plus court chemin**, ou la distance entre deux planètes est définit par les coordonnées de chacune des planètes ;
    - **Le chemin le moins dispendieux**, ou le coût entre deux planètes est égale à la distance multipliée par le prix du carburant de la planète source.
- Appliquer un scénario de conflit entre deux nations, qui empêche les chemins directs (arrête) entre deux planètes ennemies ;
- Lire des transactions à l'aide d'une entrée console et d'un menu ;
- Lire des transactions à partir d'un fichier.

4. Format des données et des fichiers textes

Les planètes :

Chaque ligne du fichier est une planète. Une planète à un nom (chaîne de caractère), des coordonnés X et Y (nombres à virgule), une population (nombre entier), une nation (chaîne de caractère) et un prix d'essence par unité de distance (nombre à virgule).

Format du fichier :

NomPlanete1 X1 Y1 POPULATION1 NATION_A PRIX1
NomPlanete2 X2 Y2 POPULATION1 NATION_B PRIX2
NomPlanete3 X3 Y3 POPULATION1 NATION_C PRIX3
…

Exemple de fichier :

Manitoba 25.2 600.87 150000 Canada 44.50
Saskatchewan 223.94 706.12 984000 Canada 40.60
Texas 441.96 52.41 85139547 USA 95.10
…

Les types de vaisseau :

Chaque ligne du fichier est un type de vaisseau. Un type de vaisseau à un modèle (chaîne de caractère) et une capacité de carburant (nombre à virgule).

Format du fichier :

Modele1 Capacite1
Modele2 Capacite2
Modele3 Capacite3
…

Exemple de fichier :

XD30-W 200
VITT-MOD-S 400
K-TURBO-500 1000
…

Les transactions à implémenter :

Votre programme doit supporter les transactions ci-dessous. Ces dernières peuvent être lu soit à la console ou à partir d'un fichier de transaction. Notez que les transactions 1 et 2 doivent être faites avant de permettre les autres transactions.

Type de transactions :

1. **#P FICHIER** // Charge un système stellaire en mémoire (fichiers de planète)
2. **#V FICHIER** // Charge les types de vaisseau en mémoire (fichier vaisseau)
3. **?1 V S D** // affiche la réponse à la question : existe-t-il une route entre les planètes S et D pour un vaisseau de type V ;
4. **?2 V S D** // Affiche le plus court chemin ainsi que la distance totale que peut parcourir le vaisseau V pour se rendre de la planète S à D.
5. **?3 V S D** // Affiche le chemin et le coût du trajet le moins dispendieux peut suivre le vaisseau V pour se rendre de la planète S à D.
6. **/ N1 N2** // Applique un scénario de conflit spatial entre les nation N1 et N2
7. **&** // Affiche toutes les planètes (info des planète), tous les types de vaisseau (info des vaisseaux) et tous les conflits entre deux nations.

Exemple de transactions :

#P PLANETE.txt
#V VAISSEAU.txt
?1 K-TURBO-500 Manitoba Texas
?2 XD30-W Saskatchewan Manitoba
/ Canada USA
?3 K-TURBO-500 Texas Saskatchewan
&

5. Structure des données

Vous êtes libre de sélectionner les structures de données et les algorithmes nécessaires au bon déroulement du devoir. **Vous devez cependant rendre votre code le plus lisible et le plus réutilisable possible.** Pour ce faire, voici quelques classes que vous devriez implémenter :

Classe vaisseau :

Définit la structure et les fonctions que doit avoir un vaisseau.

Classe planète :

Définit la structure et les fonctions que doit avoir une planète.

Classe générique d'un graphe :

Définit la structure et les fonctions de base d'un graphe.

Classe pour structurer le plateau de jeux :

Structure vous permettant de rassembler les variables et les algorithmes particuliers nécessaires à l'implémentation des fonctionnalités de votre devoir.

6. Livrables

Vous devez envoyer votre devoir à l'adresse courriel suivante : 8inf259@gmail.com
Vous devez remettre au correcteur dans un fichier .zip portant comme nom vos codes permanents (ex: ETUD01010100_ETUD02020200_TP3.zip) **seulement vos documents de sources et d'entêtes (.cpp et .h)** ainsi qu'un **fichier « lisezmoi_TP3.txt »** indiquant :

- Vos noms, prénoms et codes permanents ;
- Le système d'opération utilisé pour développer le code (Windows, Mac, Linux, etc.) ;
- L'environnement de programmation utilisée (Visual studio, Code::Block, fichier .txt, etc.);
- De préférence : La ligne de commande servant à compiler votre programme ;
- De préférence : La ligne de commande servant à exécuter votre programme ;
- **Les particularités de votre devoir qui en font un jeu.**

7. Guide de correction

Le devoir est compté sur 15 points (15% de la session). Le devoir sera évalué selon les points suivants :

- L'utilisation pertinente des structures de données vues dans le cours ;
- L'utilisation pertinente des algorithmes vues dans le cours ;
- Qualité du code quant à la structure, la lisibilité et la possibilité de réutilisation du code ;
- Bonus maximum de 2 points pour l'effort mis pour faire de votre programme un véritable jeu de gestion de ressource ;