

┌ Graphe visuel : création d'un logiciel ┐

En réalisant ce TP vous pourrez afficher des nœuds et des arêtes en utilisant LibGraph2 ou bien une autre bibliothèque, vous pouvez travailler au maximum par groupe de 2.

Positionnement du problème : un jeu multijoueurs (7 joueurs) possède 2 types de ressources : du minerai et du pétrole. Il possède 2 types d'unité de production : les simples qui ne sont accessibles que par le propriétaire originel et les avancées qui sont susceptibles d'être conquises.

Les ressources quant à elles peuvent être conquises.

Dans chaque nœud il peut y avoir au maximum 3 unités de production et maximum 5 ressources de chaque type. Les arêtes vont permettre de faire passer les ressources.

Il existe des nœuds spéciaux, appelés zone de transfert ou chaque joueur possède un certain nombre de transporteurs. Une arête reliant 2 zone de transfert est limitée au minimum du nombre de transporteur situé dans les deux zones.

Pour produire un joueur doit acheminer une ressource vers une unité de production. Peu importe la longueur du chemin. Une fois utilisée l'unité de production et la ressource ne sont plus disponibles. La difficulté est d'optimiser la production.

Le pétrole étant utilisé par ailleurs dans le jeu pour déplacer les unités, acheter des unités spéciales il est souhaitable d'en stocker à chaque tour.

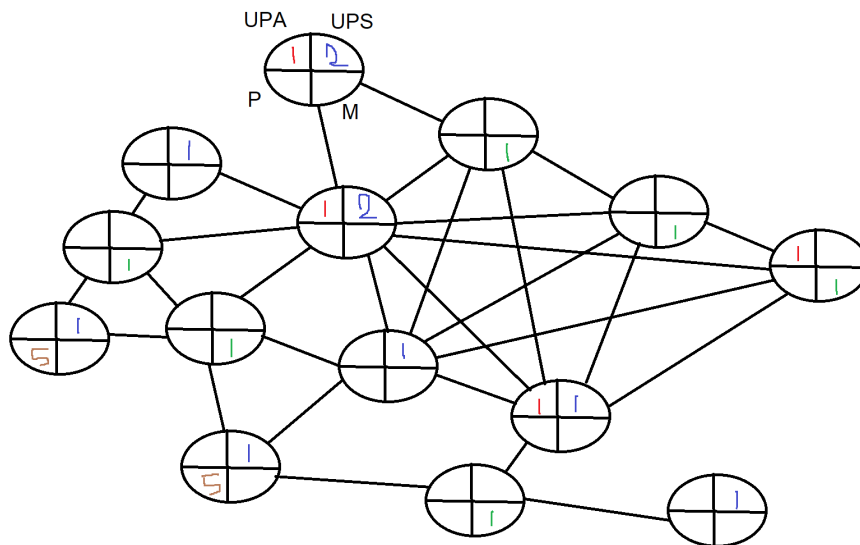
Votre travail consistera à calculer la production maximale de chaque joueur, et le nombre de pétrole disponible. En effet, un joueur peut stocker du pétrole à chaque tour pour l'utiliser plus tard.

Entre chaque tour les nœuds peuvent changer de propriétaire, les zones de transfert peuvent se voir modifier, et donc entre chaque calcul, il faut une phase de modification du graphe. (il n'y a par contre pas de destruction ni de nœud, ni d'arête).

└ Exercice 1 : Une première ébauche ▮

Le problème étant relativement compliqué, vous pouvez commencer par ne considérer qu'un seul type d'unité de production et un seul type de ressource, que les arêtes permettent de faire passer autant de ressource que l'on veut et ne vous occupez pas des zones de transfert.

Vous pouvez commencer avec un seul joueur : le joueur bleu et le graphe suivant (en fusionnant les unités de production et les ressources :



Votre programme devra afficher la production de chaque joueurs et afficher en jaune fluo les unités de production qui ne produisent pas et en vert fluo les ressources qui ne sont pas utilisées.

Vous pouvez utiliser l'optimisation de Ford-Fulkerson généralisée pour calculer le flux optimal.

└ Exercice 2 : Gestion de la lecture/écriture dans un fichier de sauvegarde ▮

Vous devez implémenter une fonction permettant la sauvegarde du graphe dans un fichier, et du coup proposer une restauration à partir d'un fichier.

└ Exercice 3 : Amélioration 1 : gestion des 2 types de ressource ▮

Dans cette partie, vous devez gérer les 2 ressources, et proposer au joueur : sa production s'en s'occuper du stock de pétrole. Vous devrez aussi gérer les zones de transfert. Je vous fournirait un graphe plus complet.

└ Exercice 4 : Amélioration 2 : gestion de la réserve en pétrole ▮

Chaque joueur devra indiquer combien de pétrole il souhaite stocker et ceci recalculera sa production. Pensez qu'au lieu de stocker il peut décider de puiser dans ses réserves stratégiques (s'il rentre un nombre négatif)