



Principe et règles du jeu

Ce jeu consiste en la gestion d'une compagnie de bus. A l'aide de vos bus, transportez vos passagers à leurs destinations pour gagner de l'argent. Utiliser cet argent pour améliorer votre réseau. L'objectif est d'atteindre une certaine somme d'argent avant vos adversaires ou d'avoir le plus d'argent à la fin du jeu.

Ce jeu constitue le projet du cours IAP 2019-2020 en FISA à l'ENSIIE. Sauf autorisation explicite de votre professeur, votre programme devra être codé en C. Le barème du projet est le suivant:

- 0 / 20 s'il reste des erreurs ou des warnings;
- 8 points sur le code (présence de commentaires, lisibilité du code, utilisation de structures

- de données et autres acquis du cours, fonctions courtes);
- 4 points pour les explications; vous devez expliquer dans un commentaire situé au début de votre fichier comment fonctionnent vos algorithmes et ce qu'ils cherchent à faire.
- 5 points si vous atteignez la ligue Bronze;
- 3 points de classement (linéaire, avec 3 points pour le premier rang et 0 pour le dernier, arrondi au demi-point supérieur).

Votre priorité est que votre programme compile et passe les 3 première ligues, nommées Bois 3, Bois 2 et Bois 1 pour atteindre la ligue Bronze. Une fois cela fait, améliorez votre code. Rien ne vous empêche de vous entraider bien entendu, mais le plagiat ne sera pas toléré. Vous devez coder seul.

Date de rendu prévisionnel : 10 janvier 2020.

Grille de jeu

Le jeu se joue sur une carte de type Grille rectangulaire; dont les coordonnées, entières, sont comprises entre $(0, 0)$ (le plus en haut à gauche de la grille) et $(10, 10)$ (le point en bas à droite de la grille). Chaque point de la grille est relié à ses 8 voisins (2 horizontaux, 2 verticaux et 4 diagonaux).

Apparitions de stations de bus

La grille contient des stations de bus où apparaissent des voyageurs et vers où se dirigent d'autres voyageurs. Au début du jeu, 3 stations sont placées au hasard. Tous les 25 tours, une nouvelle station fait son apparition jusqu'à atteindre, si le jeu ne se termine pas avant, le nombre maximum de 10 stations. Une station est identifiée par un numéro unique entre 0 et 9 ; la première station est toujours la station 0, la suivante est la 1,... Chaque station a également une capacité maximum, comprise entre 5 et 10, égale au nombre maximum de voyageurs pouvant attendre à cette station.

Apparitions des voyageurs

À chaque tour, un voyageur peut apparaître à chaque station, avec une certaine probabilité, en direction d'une autre station choisie aléatoirement (voir en bas pour les détails). Un voyageur reste sur la station jusqu'à ce qu'il trouve un bus qui le prenne ; quitte à ne jamais quitter la station si aucun bus ne vient le chercher. Dans une station, les voyageurs sont disciplinés : ils se mettent en file. Le premier arrivé est le premier à pouvoir monter dans un bus.

Comme pour les stations, un voyageur est identifié par un numéro unique. Le premier voyageur a pour numéro 0, le second est le numéro 1, ...

Les bus

Le joueur peut à chaque tour modifier son réseau (voir ci-dessous). Au fur et à mesure, il ajoute de nouveaux bus. Chaque bus est, au début du jeu, de capacité 5. Comme pour les stations et les voyageurs, chaque bus est identifié par un numéro unique dont le premier est 0.

Un bus qui se déplace en direction d'une station va pouvoir se déplacer sur la grille horizontalement, verticalement ou en diagonale. À chaque tour, le bus avance d'un cran vers la station qu'il vise ; en utilisant un plus court chemin. La vitesse d'un bus ne change pas avec le nombre de voyageurs qu'il transporte. Quand le bus arrive à sa destination, il y reste jusqu'à ce qu'on lui donne une autre destination.

Quand un bus arrive à la destination d'un voyageur, celui-ci descend et disparaît. Une fois que les voyageurs du train sont descendus, ceux qui attendent dans la station tentent de monter, dans l'ordre de la file, jusqu'à ce que le bus soit plein ou qu'il n'y ait plus de voyageurs dans la station. Le déchargement prend 1 tour pour 5 voyageurs. Ensuite le chargement prend également 1 tour pour 5 voyageurs. Même s'il n'y a que deux voyageurs qui descendent et 3 qui montent, il faudra 2 tours pour effectuer le déchargement puis le chargement. Si la destination du bus change pendant un chargement ou un déchargement, le bus stoppe ce chargement ou ce déchargement pour se diriger vers cette nouvelle station.

Argent

Le joueur démarre avec 150 pièces Bus, notées B qu'il peut utiliser pour améliorer ou gérer son réseau. Un voyageur qui arrive à destination offre 10 B au joueur.

Gestion du réseau de bus

À chaque tour, pour gérer son réseau, le joueur peut

- **Acheter un nouveau bus** pour 100 B . Attention, le joueur ne peut posséder plus de 4 bus.
- **Changer la destination d'un bus**. Il n'est pas possible de modifier la destination d'un bus qui vient d'être acheté.
- **Agrandir un bus** pour 50 B . Attention, le joueur dispose d'un nombre maximum de voitures par bus (voir plus bas). Il n'est pas possible d'agrandir un bus qui vient d'être acheté.
- **Augmenter SB , le nombre maximum de voiture de ses bus** pour 100 B . Le nombre maximum de voitures SB des bus d'un joueur est limitée. Au départ, $SB = 1$ signifiant que chaque bus du joueur ne peut avoir qu'une voiture. Le joueur peut agrandir ce nombre de 1, dans la limite de 2 fois. La taille d'un bus peut donc, au mieux, être multipliée par 2 ou par 3, passant ainsi la capacité du bus de 5 à 10 ou 15. Attention, une fois cette amélioration achetée, les bus n'acquièrent pas automatiquement une nouvelle voiture, il faut les agrandir un par un (cf ci-dessus).
- **Augmenter SP , la vitesse des bus** pour 200 B . La vitesse SP des bus d'un joueur vaut au départ 1, signifiant que chaque tour, le bus avance d'un cran vers sa destination. Le joueur peut augmenter ce nombre de 1, dans la limite de 2 fois, pour doubler ou tripler la vitesse de ses bus. Contrairement à l'amélioration précédente, l'effet de celle-ci est immédiat sur l'ensemble des bus déjà acquis.
- **Augmenter CT , le tarif des voyages** pour 100 B . Le coût CT payé par un voyageur arrivé à destination est, au départ, $CT = 10\text{B}$. Le joueur peut augmenter ce nombre de 2 B , dans la limite de 5 fois. Un voyageur peut donc payer au plus 20 B . Comme pour l'amélioration

précédente, l'effet de celle-ci est immédiat.

Fin du jeu

Le jeu s'arrête nécessairement au bout de 500 tours.

L'objectif est d'être le premier joueur à atteindre 1500\$ ou plus. Le joueur **gagne** s'il atteint cet objectif. Ce joueur est alors retiré du jeu et son score est 2000 moins le nombre de tours qu'il a mis pour atteindre cet objectif.

Le joueur **perd** si, avant d'atteindre l'objectif, il exécute une commande interdite, par exemple demander un nouveau bus alors qu'il ne dispose pas d'assez d'argent, ou s'il ne répond pas dans les temps, ou si une erreur survient lors de l'exécution de son programme. Son score est alors de 0.

Si un joueur se retrouve seul pendant la partie, le jeu s'arrête et son score est égal à 500 plus sa quantité d'argent moins le nombre de tours écoulés depuis le début du jeu.

Au bout de 500 tours, le jeu s'arrête pour les joueurs restants et leur score est égal à leur quantité d'argent.

Entrée et Sortie standards

À chaque tour, le joueur reçoit une mise à jour de l'état du jeu en *entrée standard*. Attention, certaines informations ne sont données qu'une seule fois. Par exemple la position des stations.

Au début du jeu, avant le premier tour, le joueur reçoit les entrées suivantes:

Nombre de lignes	Contenu	Nom des entrées	Détails
1	2 entiers	NJ J	Le nombre de joueurs (le joueur et ses adversaires) et le numéro du joueur

3	4 entiers	ID X Y K	Un numéro d'une des stations de départ, ID, un point (X, Y) indiquant les coordonnées de cette station, et la capacité K de la station.
---	-----------	----------	---

À chaque tour, le joueur reçoit les lignes suivantes dans l'ordre.

Nombre de lignes	Contenu	Nom des entrées	Détails
NJ	5 entiers	J M USB USP UCT	Le numéro J d'un joueur, sa quantité d'argent M et combien de fois il a amélioré les valeurs <i>SB</i> , <i>SP</i> et <i>CT</i>
1	1 entier	NS	Vaut 0 ou 1 ; indique si une nouvelle station est apparue ou non. Ce chiffre vaut donc 1 tous les 25 tours et 0 le reste du temps.
NS	4 entiers	ID X Y K	Un numéro d'une nouvelle station ID, un point (X, Y) indiquant les coordonnées de cette station, et la capacité K de la station.
1	1 entier	NBUS	Le nombre de bus en circulation.
NBUS	6 entiers	ID J X Y A S	Un numéro du bus ID, le joueur J le possédant, un point (X, Y) indiquant où se trouve le bus, un numéro de station A vers où le bus se dirige et le nombre de voitures S du bus. Remarque : le bus peut être à l'arrêt sur la station A, en train de charger ou de décharger.
1	3 entiers	NT BT DT	Le nombre de nouveaux voyageurs dans les stations, le nombre de voyageurs étant montés dans un bus au tour précédent et le nombre de voyageurs ayant atteint leur destination et quitté leur bus au tour précédent.

NT	3 entiers	IDT IDS1 IDS2	Un numéro de voyageur IDT et deux numéros de station IDS1 IDS2 où le voyageur est apparu et où il se rend.
BT	2 entiers	IDT IDB	Un numéro de voyageur IDT et un numéro de bus IDB dans lequel le voyageur est monté au dernier tour.
DT	1 entier	IDT	Un numéro de voyageur IDT ayant quitté son bus au dernier tour.

Remarque : le nombre de voyageurs n'est pas $NT + BT + DT$. Ces 3 chiffres indiquent uniquement des voyageurs ayant vécu un changement : ils sont arrivés, montés dans un bus ou descendus d'un bus. Un voyageur qui était là à une station et n'a pas bougé n'est pas réindiqué. Un voyageur dans un bus qui n'est pas descendu n'est pas réindiqué.

À chaque tour, **une fois toutes ces entrées lues**, le joueur doit envoyer en *Sortie standard* **une seule ligne se terminant par caractère de saut de ligne** contenant une ou plusieurs commandes. S'il souhaite envoyer plus qu'une commande, il peut les séparer avec un point-virgule (;), elles sont alors effectuées dans l'ordre. Les commandes possibles sont les suivantes:

- **BUS A** où A est un entier correspondant à l'identifiant d'une station. Cette commande ajoute un bus à la station A. Si le joueur ne dispose pas d'assez d'argent, si le nombre maximum de bus est déjà atteint ou si la station A n'existe pas, le joueur perd et est retiré du jeu.
- **DESTINATION B A** où B est un entier correspondant à l'identifiant d'un bus et A est un entier correspondant à l'identifiant d'une station. Cette commande change la destination du bus pour cette station. Il n'est pas possible de changer la destination d'un bus qui a été acheté durant le même tour. Si le bus B ou la station A n'existe pas, si le bus B n'appartient pas au joueur ou si le bus B a été acheté durant le même tour, le joueur perd et est retiré du jeu.
- **UPGRADE B** où B est un entier correspondant à l'identifiant d'un bus. Cette commande augmente la taille du bus B de 1. Si le joueur ne dispose pas d'assez d'argent, si le bus B a déjà atteint sa taille maximum, si le bus B n'existe pas ou s'il n'appartient pas au joueur, le

joueur perd et est retiré du jeu.

- UPDATESB qui effectue l'amélioration de *SB*. Si le joueur ne dispose pas d'assez argent ou si l'amélioration a été effectuée 3 fois, le joueur perd et est retiré du jeu.
- UPDATESP qui effectue l'amélioration de *SP*. Si le joueur ne dispose pas d'assez argent ou si l'amélioration a été effectuée 2 fois, le joueur perd et est retiré du jeu.
- UPDATECT qui effectue l'amélioration de *CT*. Si le joueur ne dispose pas d'assez argent ou si l'amélioration a été effectuée 6 fois, le joueur perd et est retiré du jeu.
- PASS, qui ne fait rien.

Voici un exemple d'entrée et un exemple de sortie.

A l'initialisation, vous pouvez recevoir

4 0

Pendant un tour, vous pouvez recevoir

```

0  253  1 0 0
1  1433 1 2 1
2  888  1 0 1
3  1326 2 5 5
1
8  2 2 8
9
0  0 1 1 3 1
1  1 4 1 2 2
2  1 6 3 4 2
3  1 0 5 3 2
4  1 7 0 6 2
5  2 9 3 2 1

```



```
6  2 5 3 1 2
7  3 7 4 0 3
8  3 4 5 7 3
3  4 2
23 1 6
24 3 4
25 5 0
17 2
7  4
13 7
14 7
11
12
```

Enfin, vous pouvez envoyer, par exemple, l'une des lignes suivantes en sortie standard

```
BUS 3; DESTINATION 2 6; PASS\n
UPDATESP\n
DESTINATION 5 0; UPDATECT; PASS ; BUS 4\n
```

Pour aller plus loin

Détails sur l'apparition des voyageurs : À chaque tour, pour chaque station non pleine, un voyageur a une probabilité $0.4 + 0.02 * (c - w)$ d'apparaître sur cette station, où c et w sont respectivement la capacité de la station et le nombre de voyageurs déjà présents dans la station. Cette probabilité est réduite au début du jeu (voir ci-après).

Détails sur la destination de voyageurs : Au début du jeu, une matrice OD (origines-destinations) est tirée au hasard entre les stations, indiquant, pour chaque station la propension des voyageurs partant de cette station à aller vers les autres stations. Cette matrice

est générée ainsi : pour chaque station, les probabilités d'aller dans les autres stations sont réparties parmi [0.40, 0.20, 0.10, 0.10, 0.05, 0.05, 0.05, 0.03, 0.02]. Ainsi, pour chaque station, il y a au moins deux stations très favorables et deux très peu favorables. Cette matrice n'est pas donnée en entrée du jeu, mais vous pouvez en calculer une estimation au fur et à mesure que le jeu évolue. **Remarque** : Lorsqu'un voyageur apparaît, il choisit une station de destination en fonction de ces probabilités. Si cette station n'est pas encore apparue alors le voyageur n'apparaît pas.

Remarque sur le chargement des bus : Si deux bus chargent en même temps à la même station, chaque voyageur choisit le bus le moins plein au moment de monter dans un bus. S'il y a plusieurs bus avec le même nombre de places libres, le voyageur choisit aléatoirement selon une loi uniforme dans quel bus il monte.

Ordre dans lequel sont calculés les tours de jeu

- Zero ou une station est ajoutée.
- Des voyageurs arrivent dans les stations.
- Les entrées sont envoyées aux joueurs.
- On récupère la sortie de chaque joueur encore en jeu.
- Si la sortie est erronée ou si aucune sortie n'est envoyée, le joueur est retiré du jeu.
- Les commandes du joueur sont exécutées.
- Pour chaque bus étant à sa destination tel qu'aucun de ses voyageurs ne souhaite descendre à cette station, le bus charge.
- Pour chaque bus étant à sa destination tel qu'au moins un de ses voyageurs souhaite descendre à cette station, le bus décharge.
- Pour tout autre bus, le bus se déplace d'un ou deux crans (selon sa vitesse) vers sa destination.
- On vérifie pour chaque joueur s'il a atteint l'objectif. Dans ce cas, il gagne et est retiré du jeu.
- S'il ne reste plus qu'un joueur ou si on a atteint le nombre maximum de tours, les joueurs restants sont retirés du jeu et la partie s'arrête.

À chaque tour, vous voyez sur l'interface graphique tout ce qui s'est passé avant de recevoir vos entrées.

Les différentes ligues et les IA précodées

Il y a 4 IAs contre lesquelles vous allez vous battre dans les ligues bois. Vous devez les vaincre pour accéder aux ligues supérieures. Pour les battre, il vous suffit d'utiliser un peu plus le potentiel que vous offre les règles. Chaque IA des niveaux supérieurs utilise plus de possibilités que les précédentes, mais aucune n'utilise toutes les possibilités.

Ligue Bois 3 : iaSNCF

Cette IA ne fait rien. Elle n'achète pas de bus, elle n'améliore pas son réseau. Elle n'envoie que la commande PASS. Pour la vaincre, vous devez simplement réussir à acheter un bus et à déposer un voyageur.

Ligue Bois 2 : iaRATP

Cette IA achète un seul bus qui fait le tour des stations par ordre d'apparition. Le bus reste sur une station 5 tours avant de passer à la station suivante. L'IA ne tient pas compte des voyageurs présents dans le bus.

Ligue Bois 1 : iaTICE, iaTransdev et iaKeolis

Vous combattez ces 3 IAs en même temps. Chacune de ces IA achète un bus dès qu'elle a assez d'argent, jusqu'à en avoir 4 ; puis elle se concentre sur l'une des 3 améliorations disponibles, sans considérer les autres. Chaque bus choisit aléatoirement au début du jeu s'il fera le tour des stations dans l'ordre ou dans l'ordre inverse. Le bus reste à sa destination 5 tours avant de se déplacer vers la destination suivante.

- iaTICE achète dès que possible 2 améliorations de *CT*.
- iaKeolis achète dès que possible 1 amélioration de *SP*.
- iaTransdev achète dès que possible 1 amélioration de *SB* puis améliore dès que possible chacun de ses bus en lui ajoutant une nouvelle voiture.

Flux de sortie et d'erreur standard

Vous devez utiliser la sortie standard pour envoyer vos commandes au jeu. Vous avez au moins 1 seconde pour envoyer votre première commande. Vous avez ensuite au moins 50 millisecondes pour envoyer les commandes suivantes. Le système utilise la classe java `ScheduledExecutorService` pour demander la sortie. Au moment où il la demande, il vous laisse 1 seconde ou 50 millisecondes selon le cas pour répondre. Cependant, plusieurs phénomènes peuvent mener à une augmentation de ce temps : premièrement, quand le jeu attend, la machine peut être occupée à faire autre chose, il se peut que la machine rende la main au jeu après 1 seconde ou 50 millisecondes, cela se joue généralement à quelques millisecondes, si la machine n'est pas trop chargée ; secondement, votre programme continue à tourner pendant que le jeu fait ses calculs pour se mettre à jour, exécuter les commandes, changer l'affichage graphique, ..., ce temps est donc offert à tous les joueurs.

Vous pouvez utiliser l'erreur standard pour envoyer des informations de débogage. Ces erreurs apparaissent à l'écran pendant le jeu. Un thread est utilisé pour récupérer votre flux d'erreur pendant le jeu, quelque soit le moment où vous l'avez envoyé. A cause de cette désynchronisation entre le jeu et le thread qui récupère l'erreur, il se peut que vos messages de debug n'apparaissent pas exactement pendant le tour où vous les avez envoyés. Ils peuvent apparaître un peu plus tard ou ne pas apparaître si votre programme s'est arrêté entre temps. Pensez à flusher votre erreur standard régulièrement si vous voulez augmenter les chances que celle-ci apparaisse au bon moment. Attention, utiliser le debug peut ralentir votre jeu. La taille de la sortie standard est limitée à 600 caractères par tour. Les sauts de lignes sont convertis en HTML, il compte donc pour 5 caractères.

Les problèmes classiques que vous risquez de rencontrer

Mal récupérer les entrées

Le jeu attend de vous que, alternativement, vous récupériez les entrées et que vous envoyiez les sorties. Si vous récupérez une entrée de trop, votre programme va attendre que le jeu vous envoie cette entrée. Pendant ce temps, le jeu va attendre que vous lui envoyiez les sorties. Donc les deux programmes sont bloqués. Vous aurez donc une erreur de TIMEOUT puisque vous n'enverrez pas les sorties à temps. Si, au contraire, vous ne récupérez pas suffisamment d'entrées, alors il y aura un décalage dans les entrées que vous recevrez. Par exemple si le jeu vous envoie 3 entiers a, b et c à chaque tour. Si vous n'en récupérez que 2, vous aurez en entrée standard a, b et c et vous récupérez a et b. L'entier c est toujours en entrée standard. Au tour suivant, le jeu vous renvoie 3 nouveaux entiers a', b' et c'. Vous récupérez 2 entiers, mais ces entiers ne seront pas a' et b'. Vous récupérerez c et a' puisque c était toujours en attente en entrée standard.

Oublier le saut de ligne

Pour chaque ligne de sortie que vous devez envoyer, n'oubliez pas de mettre un saut de ligne à la fin. Si vous ne le faites pas, vous aurez une erreur de TIMEOUT.

Attention à Python

La commande print de python effectue naturellement un saut de ligne à la fin. De plus elle place automatiquement des espaces entre tous les paramètres de la commande. Vous pouvez gérer ça avec les paramètres sep et end de la fonction.

Problèmes de flush

Il vous est conseillé de vider vos buffers de sortie et d'erreur standard à la fin de votre tour,

pour vous assurer que ces informations ne restent pas bloquées dans le buffer ; auquel cas vous risqueriez d'avoir une erreur de TIMEOUT.

- En C, utilisez `fflush(stdout)` et `fflush(stderr)`.
- En Java, utilisez `System.out.flush()` et `System.err.flush()`.
- En python3, utilisez `sys.stdout.flush()`.

Travailler en local

Vous pouvez travailler en local, en téléchargeant l'archive **minibus.tar.gz**. Vous aurez besoin de :

- De quoi faire tourner un jar Java 8 (par exemple le jdk Java 8 d'oracle ou l'openjdk8)
- Un serveur php (par exemple, wamp sous windows, la commande `php -S` sous linux)

Téléchargez et décompressez l'archive à l'endroit que vous souhaitez, tout se fait en local, aucun fichier n'est créé en dehors de ce dossier.

Pour avoir de l'aide : l'archive contient un README (plutôt adapté à la version linux).

Remarque : en local, votre sortie d'erreur n'est pas limitée en taille contrairement à la version en ligne.

Copyright © 2019, Dimitri Watel