

Manuel d'utilisation Projet JAVA : Partage de butin

EL MONTASER Osmane TD1 et VIDART Paul TD1

Sommaire

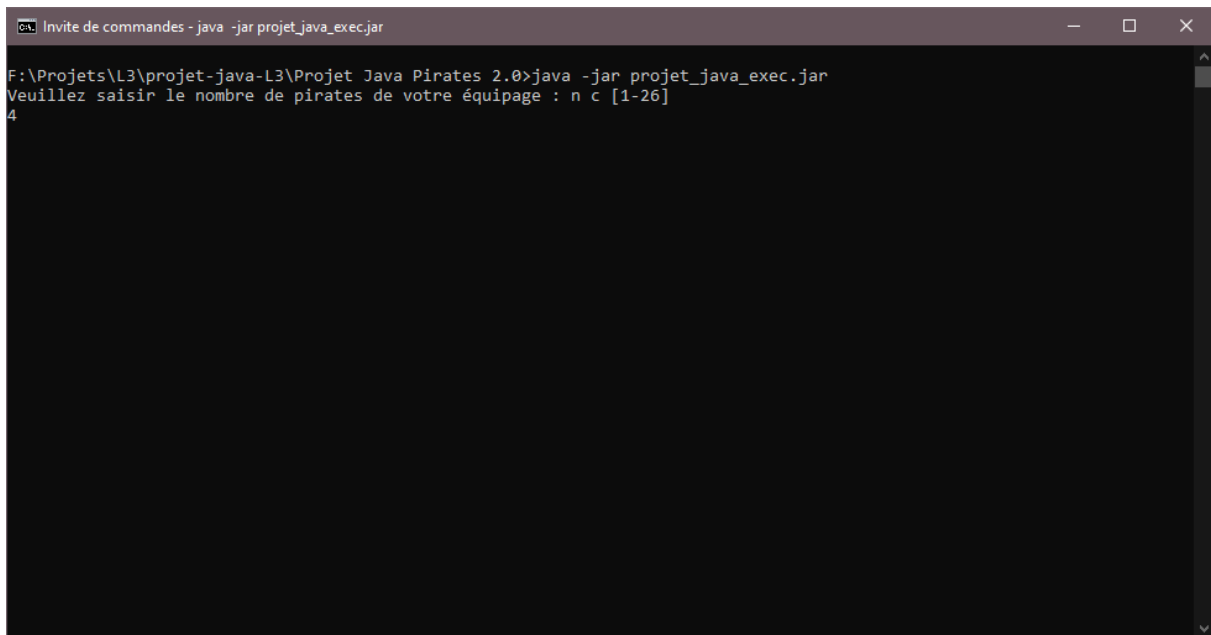
1	Instructions pour utiliser notre programme	2
1.1	Création d'un équipage de pirates	2
1.1.1	Ajout des pirates	2
1.1.2	Ajout des relations entre pirates	3
1.1.3	Ajout des préférences des pirates	5
1.2	Génération d'une solution	7
1.2.1	Échanger le butin entre 2 pirates	7
1.2.2	Calcul du coup naïf	8
2	Documentation et GitHub	9
2.1	Récupérer notre projet sur GitHub	9
2.2	Récupérer notre documentation	9

1 Instructions pour utiliser notre programme

1.1 Création d'un équipage de pirates

1.1.1 Ajout des pirates

Pour ajouter les pirates à l'équipage, il suffit de renseigner le nombre de pirates au lancement du programme en répondant à la première question :



```
CS: Invite de commandes - java -jar projet_java_exec.jar
F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
```

Figure 1: Création d'un équipage de 4 pirates

Pour l'instant, chaque pirate se voit affecter automatiquement d'un nom provenant d'une lettre de l'alphabet [A-Z]. Ce qui rend impossible la création d'un équipage de plus de 26 pirates. Si vous avez tout bien fait, le terminal devrait afficher ceci :

```
Invite de commandes - java -jar projet_java_exec.jar
F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
-
```

Figure 2: Équipage de 4 pirates généré et affiché

Vous ne pourrez pas passer à la résolution du problème tant que vous n'avez pas donné les préférences de chaque pirate.

```
Invite de commandes - java -jar projet_java_exec.jar
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
3
Les préférences du pirate A n'ont pas été affectés. Impossible de finir.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
```

Figure 3: Impossible de terminer sans donner toutes les préférences

1.1.2 Ajout des relations entre pirates

Pour ajouter une relation entre 2 pirates, il suffit de taper 1 dans le menu suivant :

```
Invite de commandes - java -jar projet_java_exec.jar
F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
1_
```

Figure 4: Sélection de l'option 1, "Ajout d'une relation"

Puis tapez la lettre correspondante aux 2 pirates, ici le pirate A n'aime pas le pirate B (et vice-versa) :

```
Invite de commandes - java -jar projet_java_exec.jar
F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
1
-- Ajout d'une relation --
Premier pirate (lettre) --> A
-- Ajout d'une relation --
Deuxième pirate (lettre) --> B_
```

Figure 5: Ajout de la relation Pirate A n'aime pas Pirate B

Appuyez sur Enter et vous retournerez au menu principal, où vous pourrez voir le résultat de l'opération :

```
Invite de commandes - java -jar projet_java_exec.jar

1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
Les pirates A (0) et B (1) ne s'aiment pas.
```

Figure 6: Résultat : Ajout de la relation Pirate A n'aime pas Pirate B

1.1.3 Ajout des préférences des pirates

Pour pouvoir calculer le coût d'une solution, il faut d'abord renseigner les préférences de chaque pirate. Vous devez taper 2 sur le menu suivant :

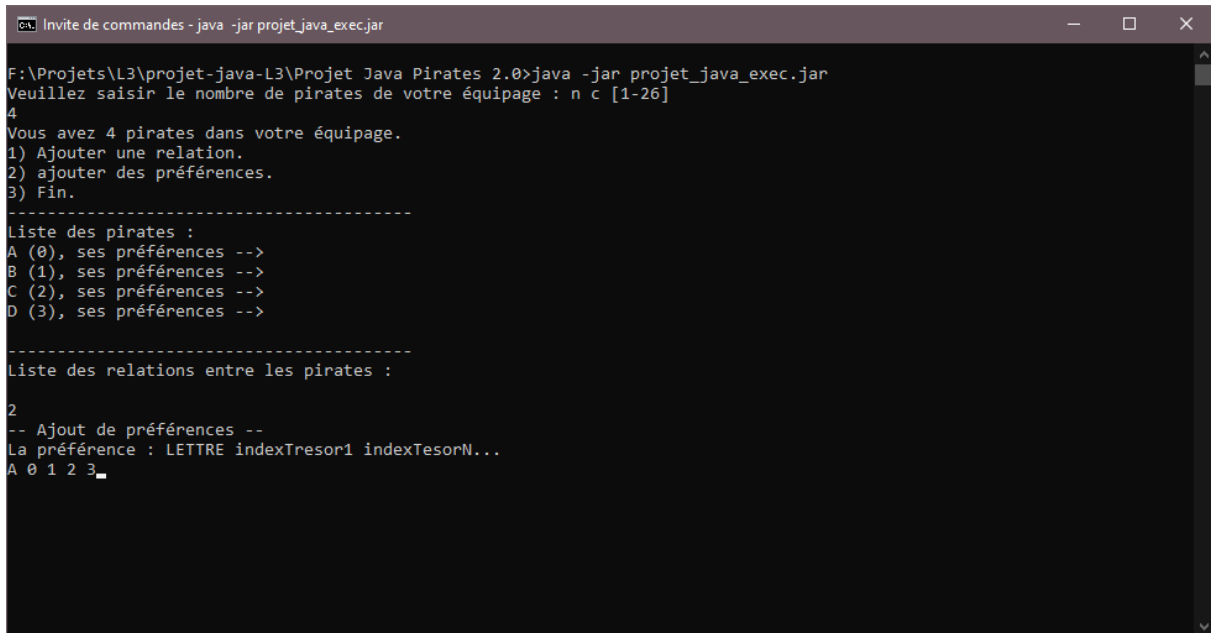
```
Invite de commandes - java -jar projet_java_exec.jar

F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
2
```

Figure 7: Sélection de l'option 2, "Ajout des préférences"

Il vous sera alors demander de taper **en une ligne** séparé par **un espace**, la lettre du

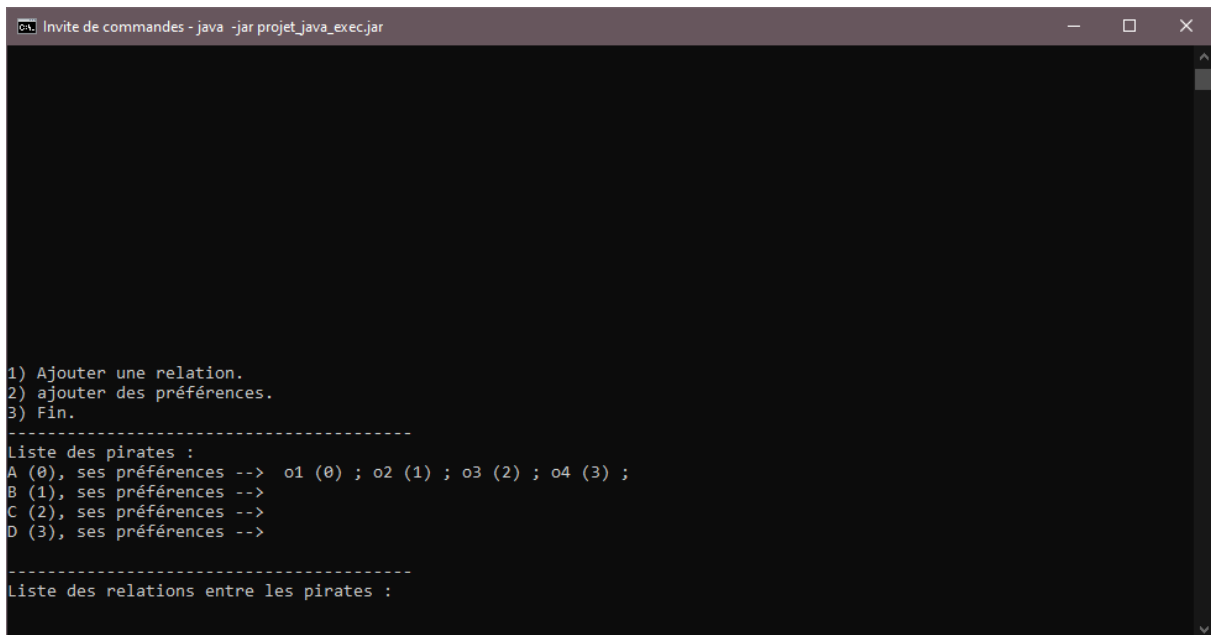
pirate, le numéro de chaque trésor ranger dans l'ordre décroissant de ses préférences. Voici un exemple pour le pirate A avec les préférences $o1 > o2 > o3 > o4$:



```
Invite de commandes - java -jar projet_java_exec.jar
F:\Projets\L3\projet-java-L3\Projet Java Pirates 2.0>java -jar projet_java_exec.jar
Veuillez saisir le nombre de pirates de votre équipage : n c [1-26]
4
Vous avez 4 pirates dans votre équipage.
1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences -->
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
2
-- Ajout de préférences --
La préférence : LETTRE indexTresor1 indexTresorN...
A 0 1 2 3_
```

Figure 8: Ajout des préférences $o1 > o2 > o3 > o4$ sur le pirate A

Appuyez sur et vous retournerez au menu principal, où vous pourrez voir le résultat de l'opération :



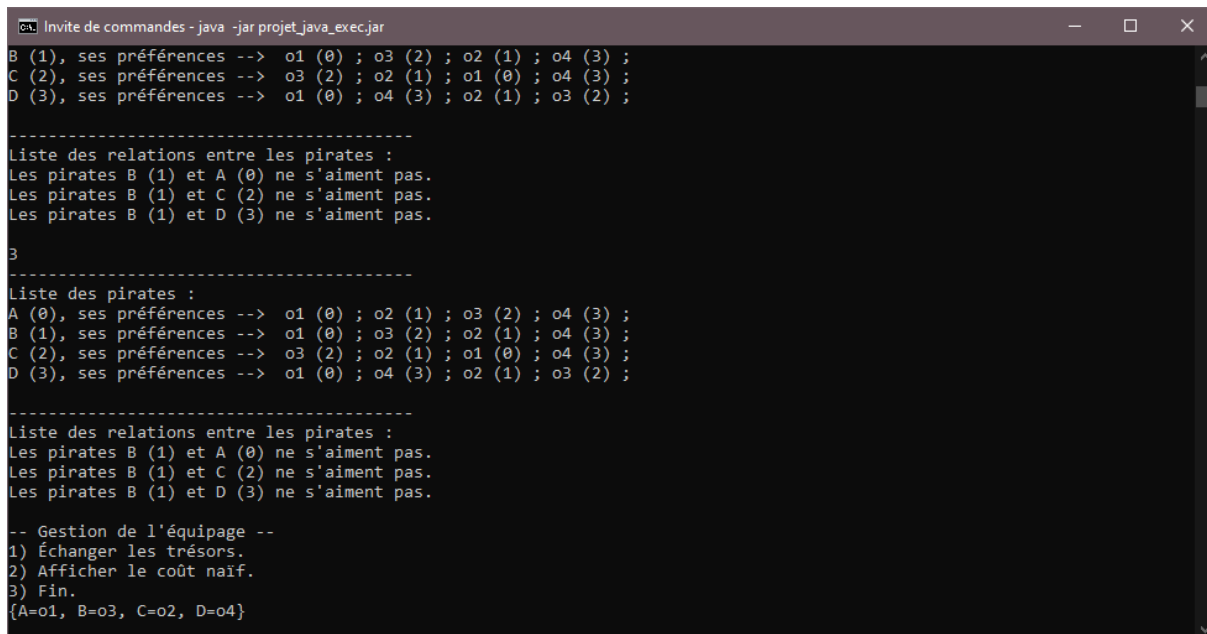
```
Invite de commandes - java -jar projet_java_exec.jar

1) Ajouter une relation.
2) ajouter des préférences.
3) Fin.
-----
Liste des pirates :
A (0), ses préférences --> o1 (0) ; o2 (1) ; o3 (2) ; o4 (3) ;
B (1), ses préférences -->
C (2), ses préférences -->
D (3), ses préférences -->
-----
Liste des relations entre les pirates :
```

Figure 9: Résultat : Ajout des préférences $o1 > o2 > o3 > o4$ sur le pirate A

1.2 Génération d'une solution

Une fois que vous avez ajouté toutes les relations entre les pirates et les préférences de chaque pirate, vous pourrez taper 3 dans le menu de création d'équipage. Le terminal devrait alors afficher ceci :



```
Invite de commandes - java -jar projet_java_exec.jar
B (1), ses préférences --> o1 (0) ; o3 (2) ; o2 (1) ; o4 (3) ;
C (2), ses préférences --> o3 (2) ; o2 (1) ; o1 (0) ; o4 (3) ;
D (3), ses préférences --> o1 (0) ; o4 (3) ; o2 (1) ; o3 (2) ;

-----
Liste des relations entre les pirates :
Les pirates B (1) et A (0) ne s'aiment pas.
Les pirates B (1) et C (2) ne s'aiment pas.
Les pirates B (1) et D (3) ne s'aiment pas.
3
-----
Liste des pirates :
A (0), ses préférences --> o1 (0) ; o2 (1) ; o3 (2) ; o4 (3) ;
B (1), ses préférences --> o1 (0) ; o3 (2) ; o2 (1) ; o4 (3) ;
C (2), ses préférences --> o3 (2) ; o2 (1) ; o1 (0) ; o4 (3) ;
D (3), ses préférences --> o1 (0) ; o4 (3) ; o2 (1) ; o3 (2) ;

-----
Liste des relations entre les pirates :
Les pirates B (1) et A (0) ne s'aiment pas.
Les pirates B (1) et C (2) ne s'aiment pas.
Les pirates B (1) et D (3) ne s'aiment pas.

-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{A=o1, B=o3, C=o2, D=o4}
```

Figure 10: Menu de gestion du butin

Vous pourrez alors voir toutes les informations sur votre équipage ainsi qu'une attribution naïve automatiquement proposée à la fin.

1.2.1 Échanger le butin entre 2 pirates

Vous pouvez échanger le butin entre les pirates en tapant 1 puis les lettres des 2 pirates entre qui échanger le butin :


```
Invite de commandes - java -jar projet_java_exec.jar
B (1), ses préférences --> o1 (0) ; o3 (2) ; o2 (1) ; o4 (3) ;
C (2), ses préférences --> o3 (2) ; o2 (1) ; o1 (0) ; o4 (3) ;
D (3), ses préférences --> o1 (0) ; o4 (3) ; o2 (1) ; o3 (2) ;

-----
Liste des relations entre les pirates :
Les pirates B (1) et A (0) ne s'aiment pas.
Les pirates B (1) et C (2) ne s'aiment pas.
Les pirates B (1) et D (3) ne s'aiment pas.

-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{A=o1, B=o3, C=o2, D=o4}
1
-- Échange de trésors --
Premier pirate (lettre) --> A
-- Échange de trésors --
Deuxième pirate (lettre) --> B
-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{C=o2, D=o4, A=o3, B=o1}
1
-- Échange de trésors --
Premier pirate (lettre) --> A
-- Échange de trésors --
Deuxième pirate (lettre) --> B
```

Figure 11: Échange des trésors entre le pirate A et le pirate B

1.2.2 Calcul du coup naïf

Pour afficher le coût de la solution naïve, il suffit de taper 2 :

```
Invite de commandes - java -jar projet_java_exec.jar
3) Fin.
{A=o1, B=o3, C=o2, D=o4}
1
-- Échange de trésors --
Premier pirate (lettre) --> A
-- Échange de trésors --
Deuxième pirate (lettre) --> B
-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{C=o2, D=o4, A=o3, B=o1}
1
-- Échange de trésors --
Premier pirate (lettre) --> A
-- Échange de trésors --
Deuxième pirate (lettre) --> B
-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{C=o2, D=o4, A=o1, B=o3}
2
Naive cost = 2.0
-- Gestion de l'équipage --
1) Échanger les trésors.
2) Afficher le coût naïf.
3) Fin.
{C=o2, D=o4, A=o1, B=o3}
```

Figure 12: Naive cost = 2.0

2 Documentation et GitHub

Pour ce projet, et afin de faciliter le développement entre nous, nous avons décidé d'utiliser la plateforme GitHub. Elle nous a permis entre autres de partager notre code source et aussi d'héberger notre documentation en ligne. (en libre accès grâce à ReadTheDocs)

2.1 Récupérer notre projet sur GitHub

Voici une commande pré-faite permettant de cloner notre dépôt git :

```
1 git clone https://github.com/Osmane-EL-MONTASER/projet-java-L3
```

Le dépôt est public, ce qui vous permet de le cloner comme vous voulez et aussi de voir les différentes versions de notre programme. Effectivement nous avons d'abord commencé une 1.0 sans implémenter les graphes, puis nous avons codé le même programme avec notre classe générique pour les graphes dans la 2.0.

2.2 Récupérer notre documentation

Pour accéder à notre documentation Javadoc, veuillez vous rendre sur ce lien :