Reversi 1

TP du module 8 - Les énumérations

Avant de démarrer ce TP, il convient d'avoir suivi les vidéos des modules 1 à 8. Son objectif est de mettre en œuvre les énumérations avec les possibilités offertes par le langage Java.

Durée estimée
4 heures

Énoncé

L'objectif est de créer un jeu de Reversi (également connu sous le nom d'Othello). Voici l'explication et les règles de ce jeu, d'après Wikipédia :

Il se joue sur un plateau unicolore de 64 cases, 8 sur 8. Les joueurs disposent de 64 pions bicolores, noirs d'un côté et blancs de l'autre. En début de partie, quatre pions sont déjà placés au centre du plateau dans la position suivante :

```
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · o ● · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
```

Chaque joueur, noir et blanc, pose l'un après l'autre un pion de sa couleur sur le plateau de jeu selon des règles précises. Le jeu s'arrête quand les deux joueurs ne peuvent plus poser de pion. On compte alors le nombre de pions. Le joueur ayant le plus grand nombre de pions de sa couleur sur le plateau a gagné.

Noir commence toujours la partie. Puis les joueurs jouent à tour de rôle, chacun étant tenu de capturer des pions adverses lors de son mouvement. Si un joueur ne peut pas capturer de pion(s) adverse(s), il est forcé de passer son tour. Si aucun des deux joueurs ne peut jouer, ou si le plateau ne comporte plus de case vide, la partie s'arrête.

La capture de pions survient lorsqu'un joueur place un de ses pions à l'extrémité d'un alignement de pions adverses contigus et dont l'autre extrémité est déjà occupée par un de ses propres pions. Les alignements considérés peuvent être une colonne, une ligne, ou une diagonale. Si le pion nouvellement placé vient fermer plusieurs alignements, il capture tous les pions adverses des lignes ainsi fermées. La capture se traduit par le retournement des pions capturés. Ces retournements n'entraînent pas d'effet de capture en cascade : seul le pion nouvellement posé est pris en compte.

Pour vous familiariser avec ce jeu, vous pouvez tester une version pour jouer en ligne telle que reversi.fr.

Indications

1 - Création du projet :

- Garder les options par défaut, notamment la séparation des fichiers sources et des fichiers compilés.

2 - Création d'une énumération `Pion` :

Donner trois valeurs possibles pour cette énumération : `LIBRE` (il n'y a pas encore de pion), `BLANC` et `NOIR`.

Ajouter un attribut `nombre` contenant le nombre de pions de cette couleur sur le plateau de jeu. Ajouter également un accesseur pour cet attribut.

Créer une méthode d'instance `getSymbole()` retournant le caractère représentant ce pion (`LIBRE` : ·, `BLANC` : o, `NOIR` : ●).

Créer une méthode d'instance `autrePion()` qui retourne le pion opposé : si cette méthode est appelé sur `BLANC`, cela retourne `NOIR`, et inversement.

Créer une méthode d'instance `gagne()` prenant en paramètre le nombre de pions qui changent de couleur suite à un coup effectué par un joueur. Cette méthode met à jour l'attribut `nombre`.

3 - Création de la classe `PlateauDeReversi` :

Le constructeur initialise le plateau de jeu en configuration de début de partie.

La méthode d'instance `afficher()` écrit sur la console les scores et dessine le plateau de jeu de la manière suivante :

```
 2 ●
 2 o
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · o ● · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
```

La méthode `tester()` prend en paramètre un pion et deux coordonnées et retourne un entier correspondant au nombre de pions qui changeraient de couleur si le joueur choisissait cette case pour poser son pion.

La méthode `peutJouer()` prend en paramètre un pion et teste s'il existe au moins une position où il est possible de le poser.

La méthode `poser()` prend en paramètre un pion et deux coordonnées. Elle positionne le pion et change la couleur de tous les pions capturés par ce coup.

Enfin, la méthode `jouer()` crée deux joueurs et les fait jouer à tour de rôle.

Exemple de début de partie :

```
Au tour de ●...
 2 ●
 2 o
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · · · · · ·
4 · · · o ● · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
ligne 3
colonne 4
Au tour de o...
 4 ●
 1 o
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · ● · · · ·
4 · · · ● ● · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
ligne 3
colonne 5
Au tour de ●...
 3 ●
 3 o
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · ● o · · ·
4 · · · ● o · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
ligne 3
colonne 6
Au tour de o...
 6 ●
 1 o
  1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · · · · ·
3 · · · ● ● ● · ·
4 · · · ● ● · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
ligne 2
colonne 5
Au tour de ●...
```

```
 4 ●
 4 o
   1 2 3 4 5 6 7 8
1 · · · · · · · ·
2 · · · · o · · ·
3 · · · ● o ● · ·
4 · · · ● o · · ·
5 · · · ● o · · ·
6 · · · · · · · ·
7 · · · · · · · ·
8 · · · · · · · ·
```

4 - Les différents types de joueurs :

Créer une interface `Joueur` avec deux méthodes :

- `jouer()` prenant en paramètre le plateau de jeu et le pion à poser et qui retourne un tableau de deux entiers contenant les coordonnées choisies pour ce pion.
- `getNom()` retournant le nom du joueur.

Créer deux classes implémentant cette interface, l'une pour faire jouer un humain et l'autre pour faire jouer l'ordinateur. Pour faire jouer l'humain, il faut demander où il souhaite positionner son pion, alors que pour l'ordinateur, il faut trouver une case où il est permis de poser son pion.

Modifier le reste du programme pour faire jouer au choix soit deux joueurs humains, soit un humain contre un ordinateur, soit deux ordinateurs.

Solution

Une solution est proposée pour ce TP sous la forme d'un PDF commenté avec le code associé. Ces éléments sont disponibles dans les ressources à télécharger.