Projet Java POO : Puissance 4 + IA

Marre de jouer à vos jeux Playstation 4 ou Switch ? Trop de 3D ? Trop de couleur ? Après avoir tout compris dans le module NFA032, vous souhaitez vous lancer dans un projet un peu fou… Le développement d'un jeu ! Vous choisissez le jeu puissance 4 en version orientée objet.

Le principe du jeu : Aligner quatre pions de même couleur horizontalement, verticalement ou en diagonale sur une grille carrée.

Avant-propos

- Veuillez lire tout une première fois avant de vous lancer.
- Les classes de votre programme doivent toutes appartenir au paquetage suivant : "net.lecnam.puissance4.[votre_nom]" (remplacer [votre_nom] par votre nom de famille).
- Des commentaires sont indispensables dans votre programme afin d'expliquer les classes et les méthodes implémentées. La génération d'une documentation à l'aide de l'outil javadoc serait un plus.
- Ne faite jamais confiance à l'utilisateur, contrôlez les entrées et gérez les erreurs.
- Votre code source devra être compressé au format zip uniquement et devra respecter le standard suivant "votre_nom.zip".
- Votre archive compressé devra m'être envoyé par mail en indiquant "[pooPuissance4] votre_nom" dans le sujet du mail.
- Je serais attentif à l'optimisation de vos programmes / nom des variables / commentaires / convention Java…
- Celui-ci doit fonctionner en sortie standard (console).
- Pas de copier / coller d'un projet existant sur Internet :) Faite le pour vous, surpassez-vous ! Vous pourriez avoir des surprises pour la note finale.

Cahier des charges

Menu

Votre programme devra intégrer un menu :

- Nouvelle partie : Votre programme lancera alors une partie avec les paramètres qui auront été modifié par le joueur
    - Deux joueurs : Lancera une partie contre deux joueurs humains
    - Joueur contre Ordinateur : Lancera une partie contre un joueur humain et la machine
    - Ordinateur contre Ordinateur : Lancera une partie contre deux ordinateurs
- Paramétrage
    - Taille de la grille de 8 à 256
    - Nombre de pions à aligner de 4 à 16
    - Niveau de l'ordinateur (3 niveaux : facile, moyen et difficile)
- Statistiques

- Quitter

Briques de base
Pour programmer ce jeu, on distinguera 3 types d'objets : des `Joueur(s)`, un `Jeu` et une `Partie`.

Un `Joueur` sera caractérisé par une couleur et par un nom.

Un `Jeu` sera une table carrée dont chaque case peut contenir un pion (c'est-à-dire une couleur), ou être vide.

Le jeu devra avoir une taille fournie lors de son initialisation (valeur par défaut : 8). Vous devrez mettre en place un paramétrage permettant de modifier la taille de la grille.

On fournira également une méthode `joueCoup` prenant un numéro de colonne et une couleur, et ajoutant le pion de cette couleur dans la colonne correspondante si c'est possible, c'est-à-dire si la colonne n'est pas pleine. Le pion doit être placé directement après le dernier pion placé sur la même colonne.

Cette méthode retournera un booléen pour indiquer si le pion a été placé ou non.

On ajoutera de plus une méthode `cherche4()` à la classe `Jeu`, qui retourne `true` s'il y a un gagnant.

La classe `Partie` sera constituée d'un joueurs ordinateur, d'un joueur humain (voir point suivant) et d'un jeu.

Les joueurs
Pour pouvoir effectivement jouer, il nous faut maintenant implémenter des joueurs.

Nous introduirons alors deux types de joueurs :

Les joueurs humains, pour lesquels il existe une méthode `joue` qui consiste à afficher le `Jeu` (faire le nécessaire), demander à l'écran le numéro de colonne à jouer jusqu'à ce que le coup soit valide, puis le jouer effectivement sur le Jeu.

L'ordinateur, pour lequel dans cet exercice il existe une méthode `joue` très simple : jouer en la première position possible rencontrée.
On informera l'utilisateur du coup joué par un message à l'écran.
Vous devrez implémenter 3 niveaux de difficultés pour l'ordinateur :
- Facile : Le joueur gagne facilement contre l'ordinateur
- Moyen : L'ordinateur gagne dans 50% du temps
- Difficile : L'ordinateur gagne à tous les coups

La partie
Testez votre programme en créant une partie dans la méthode `main`, avec un Joueur

humain et un ordinateur (qui commence).

Pour cela la classe `Partie` devra avoir une méthode `joue` qui fait jouer les joueurs tour à tour et vérifie à chaque fois s'il y a un gagnant ou si le jeu est plein et affiche le résultat en conséquence.

Statistiques
Un certain nombre de statistiques sont à mettre en place dans le menu qui porte le même nom.

- Nombre de parties joué
- Nombre de victoire pour les joueurs et l'ordinateur dans les 3 modes de difficultés différentes

Les données relatives aux statistiques doivent être enregistré dans un fichier prévu à cette effet.

Exemple du déroulement optimal

Dans l'exemple ci-dessous 12345678 indiquent des numéros de colonnes.

```
Entrez votre nom:
Thomas
--
Le programme a joué en 1
```

```
B
--------
12345678
Thomas, c'est à vous. Entrez un numéro de colonne  (entre 1 et 8)
:
1
Le programme a joué en 1
```

```
B
R
B
```

```
--------
12345678
Thomas, c’est à vous. Entrez un numéro de colonne  (entre 1 et 8)
:
2
Le programme a joué en 1




B
B
R
BR
--------
12345678
...
```