

Objectif du projet : Créer une application en Java offrant la possibilité de jouer à plusieurs jeux à deux joueurs.

Le projet est à réaliser par trinôme. Il suivra un développement itératif et incrémental. Les itérations seront de deux semaines, les butées temporelles correspondront aux séances de TP au cours desquelles vous devrez faire une démonstration du fonctionnement de votre projet. Il est important d'avoir un projet opérationnel à présenter à chaque séance même s'il ne couvre pas toutes les fonctionnalités.

Ce sujet détaille le travail demandé pour la première itération : développer une application permettant de jouer au jeu Othello à deux joueurs.

L'itération suivante sera consacrée à développer une application permettant de jouer au jeu Othello à deux joueurs ou contre l'ordinateur.

Les deux itérations suivantes porteront sur l'intégration d'un autre jeu.

Les informations vous seront données au fur et à mesure de l'avancement du projet.

Tout au long du projet, vous devrez toujours respecter une architecture MVC.

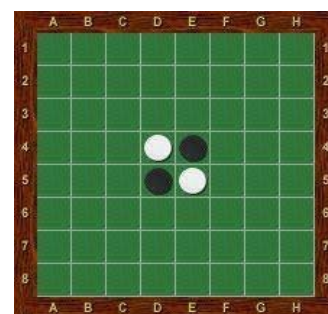
Itération 1 :

Vous devez développer une application Java permettant à deux joueurs humains de jouer au jeu Othello en mode console.

Othello est basé sur le jeu Reversi qui a été inventé en 1883 par l'Anglais Lewis Waterman.

C'est un jeu de stratégie à deux joueurs : Noir et Blanc. Il se joue sur un damier vert de 64 cases, 8 sur 8.

Ces joueurs disposent de 64 pions bicolores, noirs d'un côté et blancs de l'autre. Par commodité, chaque joueur a devant lui 30 pions (2 de sa couleur sont déjà placés sur le plateau au début de la partie, comme le montre le schéma ci-contre).



But du jeu

Avoir plus de pions de sa couleur que l'adversaire à la fin de la partie.

Celle-ci s'achève quand aucun des deux joueurs ne peut plus jouer de coup légal.

La pose d'un pion

A son tour de jeu, le joueur doit poser un pion de sa couleur sur une case vide du plateau, adjacente à un pion adverse. En posant son pion, il doit encadrer un ou plusieurs pions adverses entre le pion qu'il pose et un pion de sa couleur déjà placé sur le plateau, que ce soit sur une ligne horizontale, verticale ou diagonale.

Les pions encadrés sont alors retournés en pions de couleur inverse. Les pions ne sont jamais retirés du plateau, ni déplacés d'une case à l'autre.

Si un joueur ne peut pas poser de pions, il passe son tour.

Déroulement du jeu

Au démarrage, le programme demande le nom du joueur 1. Le joueur 1 saisit son nom. Le programme demande le nom du joueur 2. Le joueur 2 saisit son nom.

La partie peut alors commencer. Le joueur 1 qui a les noirs commence.

Avant chaque coup, le programme affiche l'état de la partie et invite le joueur courant à jouer son coup en rappelant son nom et la forme d'un coup.

Par exemple, on pourra afficher : " Dupont à vous de jouer. Saisir une ligne entre 1 et 8 suivie d'une lettre entre A et H (ex : 3 D) ou P pour passer son tour. "

Si le coup est invalide (ligne invalide ou colonne invalide ou impossible de passer son tour ou impossible de jouer sur la case indiquée), on en informe le joueur et on lui propose de rejouer.

Si le coup est valide, le programme applique le coup et teste si la partie est finie. Si elle n'est pas finie, c'est à l'autre joueur de jouer sinon le programme détermine le vainqueur, affiche son nom et le nombre de pions de chacun des joueurs, puis propose de rejouer une partie. S'il n'y a pas de vainqueur, le programme affiche "ex aequo".

Si les joueurs décident de rejouer une partie, le programme redémarre une partie. S'ils décident d'arrêter, le programme affiche le nombre de parties gagnées par chacun des joueurs et le nom du vainqueur ou "ex aequo" si les deux joueurs ont remporté le même nombre de parties.

Architecture Modèle Vue Contrôleur

Pour mettre en place l'architecture MVC, vous devrez créer les 3 packages : `vue`, `controleur` et `modele`.

Le package `vue` contiendra une classe `Ihm`.

Le package `controleur` une classe `Controleur`

Le package `modele` contiendra les classes qui découleront de votre analyse et conception de projet.

La classe Ihm

Cette classe gère toutes les interactions avec l'utilisateur. Elle se chargera d'afficher les informations, de récupérer les données saisies par l'utilisateur à l'aide de la classe Scanner et de les renvoyer au contrôleur.

La classe Controleur

Cette classe gère le déroulement du jeu, c'est à dire l'enchaînement des parties successives jusqu'à ce que les joueurs décident d'arrêter. A sa création, le `controleur` reçoit un objet de type `Ihm`. Il communique avec les objets du modèle pour la gestion de chaque partie.

A vous de prévoir les méthodes que doit fournir le controleur.

On ajoute un package `main` contenant une classe `Main` qui contient le `main`. Voici le code de la méthode `main` :

```
public static void main(String[] args) {
    Ihm ihm = new Ihm();
    Controleur controleur=new Controleur(ihm);
    controleur.jouer();
}
```

On notera que seules les méthodes de l'objet de type `Ihm` permettront d'interagir avec l'utilisateur.

En dehors de cette classe, il n'y aura aucune utilisation de `System.out.println` ou d'un objet de type `Scanner sc = new Scanner(System.in)`.

Pour faciliter le test de votre application, vous définirez un damier d'une taille 8 correspondant au nombre de lignes et de colonnes mais qu'on pourra, par exemple, réduire à 4 si besoin.

Pour l'affichage du damier, vous ferez apparaître les lettres de A à H en haut et les lignes de 1 à 8 à gauche.

Les pions blancs pourront être obtenus en utilisant la chaîne de caractères `"\u26AA"`.

Les pions noirs pourront être obtenus en utilisant la chaîne de caractères `"\u26AB"`.

Les cases vertes vides pourront être obtenues en utilisant la chaîne de caractères `"\uD83D\uDFE9"`.

Travail à rendre avant le 06/03 :

Vous déposerez sur CELENE une archive au format zip contenant l'analyse de l'application, un modèle de conception ainsi que le code de votre projet dont vous ferez la démonstration pendant la première séance de TP

Pour l'analyse, vous fournirez un diagramme de cas d'utilisation avec la rédaction détaillée du ou des scénarios.

Pour le modèle de conception, vous fournirez le diagramme de classe de vos classes logicielles avec les attributs, les opérations et les associations entre les classes.