

Devoir

Escampe

Pour pouvoir lancer le serveur de jeu en réseau, vous avez besoin d'intégrer votre propre projet avec les 4 classes du paquetage `escampe` que vous trouverez dans le même répertoire que ce fichier. Vous devez télécharger ces sources java et les mettre dans un répertoire (qui fait office de paquetage) appelé `escampe`. Ces quatres classes sont :

- `escampe.Applet` est la classe permettant de gérer une applet permettant un affichage graphique du plateau de jeu. Attention, lors des parties, le dernier coup n'est pas affiché.
- `escampe.ClientJeu` est le client réseau permettant de faire circuler vos coups en messages vers (et de) l'arbitre, via une connection réseau. La communication avec votre IA se fait via l'interface `escampe.IJoueur`.
- `escampe.IJoueur` est l'interface définissant les méthodes que vous devez implanter pour entrer dans le tournoi, et permettre à `ClientJeu` de communiquer avec vous. Vous devez étudier cette interface et l'implanter avec votre propre classe IA.
- `escampe.Solo` est une classe contenant une méthode `main` et mettant en relation deux classes `IJoueur` pour qu'elles jouent l'une contre l'autre. Cela vous permettra de mettre au point votre IA en conditions presque réelles du tournoi (attention : sans vérifier le dépassement de temps, ni la légalité des coups), en chargeant dynamiquement les classes que vous pourrez lui donner en paramètres.

L'archive JAR contenant les classes (compilées) du serveur de jeu en réseau est disponible que vous trouvez dans le même répertoire que ce fichier. Pour lancer le serveur, il faut lancer la commande suivante, après avoir récupéré le `.jar` comme il se doit :

```
java -cp escampeobf.jar escampe.ServeurJeu 1234 1
```

Cette commande permet de lancer un `ServeurJeu` sur la machine courante. Vous noterez que le serveur est indépendant, à l'intérieur de son propre jar et vous n'avez pas à le placer avec votre propre projet. Sur cet exemple, le serveur attend des connections sur le port de la machine locale, numéro 1234.

Une fois que le serveur est lancé, il faut lancer deux `ClientJeu`, chacun avec une classe de `IJoueur` à charger. Pour cela, il faut obligatoirement que `ClientJeu` soit compilé avec la même version Java que votre `IJoueur`. Comme chaque `ClientJeu` communique avec le serveur via le réseau, chaque joueur pourra être compilé avec sa propre version de Java. Vous devrez **impérativement** compiler ces fichiers avec votre projet pour entrer dans le tournoi.

Pour lancer un `ClientJeu`, en supposant par exemple qu'on veuille lancer la classe `JoueurSuperFort` (qui implante la classe `IJoueur`), et en supposant que tout le projet est dans un paquetage nommé `escampe` il faut compiler le tout et lancer :

```
java -cp . escampe.ClientJeu escampe.JoueurSuperFort localhost 1234
```

Le `ClientJeu` va charger la classe `escampe.JoueurSuperFort` (que vous avez créé) et se connecter sur le serveur qui devrait être sur la machine localhost, sur le port 1234. Vous noterez qu'il faut faire cela deux fois, de manière à connecter deux joueurs. Au total, vous aurez donc 3 terminaux shell qui s'agiteront! Vous pouvez aussi changer à loisir les classes dont vous avez les sources, dans la mesure où seuls les coups circulent sur le réseau, sous forme de chaines de caractères (et non des objets java).

Nous vous fournissons aussi, dans l'archive jar donnée ci-dessus, un joueur aléatoire pour tester vos IA. Pour lancer ce joueur aléatoire, il faut lancer :

```
java -cp escampeobf.jar escampe.ClientJeu escampe.JoueurAleatoire localhost 1234
```

Pour lancer le joueur humain :

```
java -cp escampeobf.jar escampe.ClientJeu escampe.JoueurHumain localhost 1234
```

La durée maximale sera 5 minutes par partie et par joueur.

A rendre sur eCampus :

- Le rapport (cf. indications dans le sujet)
- Les sources de votre projet
- Un `.jar` executable `Nom1Nom2.jar`