

# Projet APO

Mathieu Lefort

18 décembre 2018

## 1 Modalités pratiques

Ce TP est à faire en trinôme et est à rendre pour le 27 janvier 2019.

## 2 Évaluation

Vous serez évalués sur les points suivants :

- Rapport (1 point)
- Conception (7 points)
- Code (6 points)
- Documentation (2 points)
- Extensions (à intégrer dans les différents parties) (? points suivant vos choix)

### 2.1 Rapport

Vous rendrez un rapport détaillant votre méthodologie de travail, la répartition des tâches entre les différents membres du trinôme et une évaluation du temps passé par chacun sur le projet.

### 2.2 Conception

Vous utiliserez les différents types de diagrammes vu en TD à bon escient (i.e. permettant à quelqu'un d'extérieur au projet de savoir de quoi il est question et de coder le programme). Pour rappel, pour le diagramme de cas d'utilisations, chaque cas d'utilisation doit être décrit par un texte.

Votre conception devra être modulaire et donc permettre une intégration sans trop d'efforts des différentes extensions possibles. Il ne vous est pas demandé de modéliser les extensions non faites, mais d'intégrer leur éventuelle insertion dans votre réflexion lors de votre conception.

Les différents choix (majeurs) de votre conception devront être justifiés dans le rapport.

### 2.3 Code

Le code devra être clair (indentation, noms de variables pertinentes, ...) et être autant que possible efficace. Il devra également être en accord avec la conception. Penser à bien vérifier et traiter (avec des exceptions) les cas d'erreurs dans vos méthodes.

Penser à utiliser au maximum les packages pour structurer vos fichiers/classes.

### 2.4 Documentation

Vous générerez la javadoc de votre code (documentation des classes, méthodes, paramètres, ...). Vous penserez également à commenter votre code là où c'est utile.

## 2.5 Extensions

Plusieurs extensions vous sont proposées. À vous de voir celles qui vous intéressent le plus. Il est tout à fait possible également de proposer (et d'implémenter) vos propres extensions. Suivant le nombre d'extensions que vous choisirez, votre note maximale potentielle peut dépasser 20. En pratique votre note sera plafonnée à 20 pour le projet, les points supplémentaires pourront être transférés à votre note de CC avec un barème à définir. L'idée des extensions est surtout une opportunité de pousser l'exploration de Java plus loin si vous le souhaitez.

## 3 Sujet

Dans le jeu du puissance 4, le plateau de jeu est une grille (verticale) qui se compose de 7 colonnes, chacune constituée de 6 lignes. À tour de rôle, chacun des 2 joueurs, va jouer un de ses pions dans une des colonnes et son pion va alors tomber au plus bas de la colonne, arrêté par le fond ou un autre pion. Voici un exemple où '1' représente les pions du joueur 1, '2' représente les pions du joueur 2 et 'X' représente les positions où le prochain joueur pourra mettre un de ses pions.

			X			
		X	2	X		
		2	1	1	X	
X	X	1	2	1	2	X

Le jeu s'arrête dès que 4 pions sont alignés (d'où le nom du jeu), que ce soit horizontalement, verticalement ou en diagonale. Voici des exemples de configurations gagnantes pour le joueur 1 :

			1			
		1	2	2		
	1	2	1	1	2	
1	2	1	2	1	2	2

		1				
		1	2	2		
	2	1	1	1	2	
1	2	1	2	1	2	2

		2	1	2		
	2	1	1	1	1	
1	2	2	2	1	2	1

Si personne n'a gagné une fois la grille remplie, alors il s'agit d'un match nul.

## 4 Tronc commun

Vous coderez un jeu du puissance 4. Votre programme doit

- permettre d'afficher l'état courant de la grille, sous le format textuel suivant, les pions du premier (respectivement second) joueur seront représentés par 'X' (respectivement 'O') :

```
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| |O| | | | |
| |X|O|X| | |
```

-----  
1 2 3 4 5 6 7

- demander au prochain joueur le numéro de la colonne dans la laquelle il souhaite jouer et effectuer le coup (ou le refuser si le coup n'est pas possible).
- afficher un message indiquant la fin de partie le cas échéant puis proposer soit de quitter le jeu, soit de refaire une partie

## 5 Extensions possibles

### 5.1 Environnement de travail (1 point)

Vous mettrez en place un git (accessible) pour faire le suivant de versions de votre projet.

### 5.2 Installation (2 points)

Vous fournirez un fichier build.xml (regarder <https://ant.apache.org/>) permettant de compiler votre code, de générer la javadoc et de supprimer l'installation (fichiers compilés et documentation). Si vous faites également une extension demandant l'utilisation d'un framework hors JRE, le fichier devra également s'occuper de l'installation du framework le cas échéant.

### 5.3 Personnalisation (2 points)

Dans cette extension, vous ajouterez les fonctionnalités suivantes (par exemple en les proposant dans un menu initial ou en le paramétrant dans des fichiers de configuration) :

- chaque joueur doit pouvoir choisir le symbole qui le représente
- il est possible de choisir sa propre taille de grille ainsi que le nombre de pions à aligner pour gagner
- chaque joueur doit pouvoir assigner ses propres touches pour désigner les colonnes du jeu
- l'affichage peut soit être la suite des grilles (affichage par défaut), soit uniquement la grille courante (il faut donc effacer le contenu de l'invite de commandes à chaque fois)
- le jeu se joue au meilleur de N parties

### 5.4 Rejouabilité (2 points)

- la partie peut être sauvegardée pour être reprise plus tard
- le coup précédent pourra être annulé
- une fois la partie terminée, le programme permettra de l'enregistrer pour la rejouer plus tard. Il sera alors possible de la reprendre à n'importe quel moment.

### 5.5 Tests unitaires (2 points)

Vous implémenterez des tests unitaires des méthodes principales de votre programme (regarder <https://junit.org/junit5/>).

### 5.6 Interface graphique (4 points)

En plus d'une interface textuelle, une interface graphique (pour l'affichage de la grille mais aussi pour le choix de la colonne) sera disponible.

### 5.7 Intelligence artificielle (4 points)

En plus de fournir le mode humain contre humain, vous proposerez le mode humain contre ordinateur. Afin de faire une IA capable de jouer, vous pouvez regarder les algorithmes MCTS ([https://en.wikipedia.org/wiki/Monte\\_Carlo\\_tree\\_search](https://en.wikipedia.org/wiki/Monte_Carlo_tree_search)). Plus d'informations sur demande.

### 5.8 Partie distante (4 points)

Vous proposerez lors du lancement de la partie de faire une partie avec une autre personne sur un autre ordinateur. Le second joueur devra pouvoir se connecter à la partie lancée en précisant l'adresse IP de l'hôte. Votre programme devra également gérer les cas de déconnexion possibles.