

# Projet APO

## Jeux de Société au Tour par Tour

Ahmad SHAHWAN

4 décembre 2025

## 1 Modalités pratiques

Ce TP est à faire en trinôme et à rendre avant le 6 janvier 2026 à minuit.

### 1.1 Livrables

Votre travail doit être rendu sous la forme d'un dépôt Git en suivant le lien dédié au TP sur GitHub Classroom. Seuls les dépôts créés via ce lien seront évalués et notés. Aucun autre dépôt personnel, privé ou public, ne sera examiné.

Votre dépôt doit contenir les éléments suivants :

- Un rapport au format Markdown ou PDF. Voir la section suivante.
- Le code source du projet dans un répertoire dédié.
- La Javadoc générée dans un répertoire dédié.
- Les diagrammes UML dans un répertoire dédié.

### 1.2 Rapport

Le rapport contiendra au maximum 1600 mots. Les noms et prénoms des membres du trinôme doivent apparaître, de manière claire, au début du rapport, faute de quoi votre travail ne sera pas évalué.

Le rapport présentera votre méthodologie, les diagrammes UML et les choix de conception. Le rapport doit également présenter la structure de votre dépôt Git.

### 1.3 Démonstration et soutenance

Vous devrez présenter votre travail auprès de l'enseignant responsable du projet le 7 janvier 2026 entre 14h00 et 17h15. La soutenance durera 9 minutes pour chaque trinôme, dont 5 minutes de démonstration par les membres du trinôme et 4 minutes de questions-réponses. L'heure précise de passage pour chaque trinôme sera communiquée ultérieurement.

## 2 Évaluation

Vous serez évalués selon les critères suivants :

- Méthodologie (1 point)
- Démonstration (2 points)
- Conception (7 points)
- Code (8 points)
- Documentation (2 points)
- Extensions optionnelles (points bonus variables)

Deux extensions fonctionnelles sont obligatoires et contribuent aux cinq premiers critères. La réalisation d'autres extensions vous permet d'obtenir des points supplémentaires. Toutefois, la note finale reste plafonnée à 20.

### 2.1 Méthodologie

Vous expliquerez dans votre rapport la méthodologie de travail : articulation entre conception et codage, planification temporelle, organisation des tâches, etc. Vous expliquerez aussi la répartition des tâches entre les membres du trinôme.

## 2.2 Conception

Vous utiliserez les différents types de diagrammes vus en TDs et en cours à bon escient. Votre modèle doit permettre à une personne extérieure au projet de le comprendre et de le coder. Pour rappel, pour le diagramme de cas d'utilisation, chaque cas d'utilisation doit être décrit par un texte.

Votre conception devra être modulaire et permettre une intégration sans difficulté des différentes extensions possibles. Les choix majeurs de votre conception devront être justifiés.

## 2.3 Code

Vous réaliserez votre conception en Java. Vous avez la liberté de choisir l'une des versions LTS suivantes : 11, 17, 21 ou 25. Votre choix de version Java devra être mentionné explicitement dans votre rapport.

Le code devra être clair : indentation cohérente, conventions de nommage appropriées et noms pertinents pour les différents éléments de code. Il devra également être efficace et conforme à votre conception. Vous devrez traiter les cas d'erreur dans votre code. Pensez à utiliser les paquetages pour structurer votre code.

## 2.4 Documentation

Vous générerez la Javadoc de votre code : documentation des classes, interfaces, méthodes et paramètres. Pensez également à commenter votre code là où vous le jugez utile.

# 3 Sujet

L'objectif de ce projet est de créer un moteur de jeu pour les jeux de société à deux joueurs, au tour par tour. Ensuite, il faudra réaliser les jeux de tic-tac-toe et de puissance-quatre à l'aide de ce moteur.

## 3.1 Le moteur de jeu

L'objectif de ce module logiciel est d'abstraire les fonctionnalités communes aux jeux de société à tour de rôle. Le moteur de jeu assure le déroulement d'une partie, notamment les fonctions suivantes :

1. Faire passer le tour d'un joueur à l'autre.
2. Lire l'action d'un joueur.
3. Afficher l'état de la partie.

### 3.1.1 Déroulement d'une partie

À son tour, le joueur est invité à effectuer une action en fournissant une chaîne de caractères qui définit son coup. À ce moment-là, l'état du jeu est évalué. L'évaluation de l'état du jeu peut mener à l'un des trois résultats suivants :

1. La partie est nulle.
2. Un gagnant a été trouvé.
3. La partie est toujours en cours.

Dans tous les cas, une représentation de l'état du jeu est affichée pour tous les joueurs. Dans les deux premiers cas, la partie s'arrête. Dans le dernier cas, la partie continue et le joueur suivant est invité à jouer.

### 3.1.2 Lecture d'une action

Les actions possibles varient selon le jeu. Chaque action, généralement une coordonnée sur un tableau, est définie par une chaîne de caractères. L'utilisateur est invité à fournir une action valide lors de son tour via le terminal. Tant que la chaîne de caractères fournie ne peut pas être interprétée ou ne correspond pas à une action valide, l'utilisateur est invité à renouveler sa tentative.

### 3.1.3 Affichage de l'état de la partie

L'état de la partie, généralement un tableau, est affiché à la sortie du terminal au début du jeu et après chaque tour.

### 3.1.4 Modalité du jeu

Par défaut, les jeux implémentés doivent pouvoir être joués sur un terminal. Une extension permettra de jouer en réseau. Votre conception doit anticiper cette extension.

Chaque modalité doit donc être considérée comme une variante du moteur de jeu : jeu en terminal, jeu en réseau par connexion directe, ou jeu en réseau avec serveur centralisé.

## 3.2 La jouabilité

Alors que le moteur de jeu doit rester le même, la jouabilité (ou *gameplay*) varie selon chaque jeu.

Pour les jeux de morpion, dont le tic-tac-toe et le puissance-quatre font partie, un tableau, généralement bidimensionnel, représente l'état de la partie. Au départ, le tableau est vide. À son tour, chacun des deux joueurs place un pion selon certaines règles. Le joueur gagne s'il parvient à aligner un certain nombre de ses pions, verticalement, horizontalement ou diagonalement. La taille d'un alignement gagnant est définie par le jeu. La partie est considérée nulle lorsque le tableau est rempli sans aucun alignement gagnant.

### 3.2.1 Règles spécifiques au tic-tac-toe

Le tic-tac-toe standard se joue sur un tableau de  $3 \times 3$ . Un alignement gagnant consiste en 3 pions du même joueur. À son tour, le joueur peut placer son pion dans n'importe quelle case libre.

### 3.2.2 Règles spécifiques au puissance-quatre

Le tableau du puissance-quatre standard comporte 6 lignes et 7 colonnes. La longueur d'un alignement gagnant est de 4 pions du même joueur. Les pions sont empilés dans des colonnes verticales. À son tour, le joueur choisit une colonne non pleine et place son pion en haut de la pile.

## 3.3 Déroulement, actions et affichages par jeu

Les sections suivantes décrivent le déroulement, les actions et les affichages attendus pour chaque jeu.

### 3.3.1 Tic-tac-toe

Au départ d'une partie, le tableau du jeu, actuellement vide, s'affiche à la sortie du terminal. Pour chaque case inoccupée un libellé variant de 1 à 9 est affiché à l'intérieur. Le libellé 1 correspond à la case dans le coin en haut à gauche, et le libellé 9 correspond à la case dans le coin diagonalement opposé. Ces libellés numériques représentent les actions possibles à chaque stade de la partie. Une case déjà occupée par le pion d'un joueur contiendra le libellé représentant ce joueur, traditionnellement X ou O.

	1	2	3	
	4	5	6	
	7	8	9	

FIGURE 1 – Affichage du tableau de jeu du tic-tac-toe en configuration initiale.

À son tour, le joueur est invité à choisir l'emplacement de son prochain pion en saisissant le libellé de la case ciblée. Si le libellé n'existe pas, ou si la case est occupée, le joueur est invité à effectuer un nouveau choix.

Lorsqu'un choix valide est effectué, le tableau est mis à jour avec le pion du joueur en cours placé dans la case choisie. À ce moment-là, une vérification est réalisée pour déterminer si le joueur actuel a gagné, si la partie est nulle, ou si le jeu se poursuit.

	X	2	3	
	4	5	6	
	7	8	9	

FIGURE 2 – Affichage du tableau de jeu du tic-tac-toe suite au placement du pion du joueur X dans la case à gauche en haut.

Dans tous les cas, le tableau est affiché aux deux joueurs pour inspection. Si un joueur est déclaré gagnant, la configuration gagnante, à savoir un alignement de trois cases, est mise en évidence sur le tableau. Si la partie est nulle, un message adéquat accompagnera le tableau plein. Sinon, le tour passe à l'autre joueur. C'est alors à lui de faire son choix du prochain placement.

	(X)	2	0	
	4	(X)	0	
	7	8	(X)	

FIGURE 3 – Affichage du tableau de jeu du tic-tac-toe lorsque le joueur X remporte la partie, avec l’alignement gagnant mis en évidence.

### 3.3.2 Puissance-quatre

Un tableau bidimensionnel représentant le plateau de jeu s’affiche sur le terminal au départ de la partie et entre chaque tour. Le tableau affiché comporte 7 lignes au total : 6 lignes pour le plateau de jeu et 1 ligne pour les numéros de colonnes, numérotées de 1 à 7. La ligne du bas est réservée à l’affichage des numéros de colonnes. Ces libellés représentent les actions possibles. Les six lignes supérieures constituent le plateau de jeu proprement dit.

Les cases inoccupées sont vides et n’affichent aucun contenu. Une case déjà occupée par le pion d’un joueur affiche le symbole représentant ce joueur. Par extension, nous utilisons les symboles X et O.

	1	2	3	4	5	6	7	

FIGURE 4 – Affichage du tableau de jeu du puissance-quatre en configuration initiale.

À son tour, le joueur est invité à choisir l’emplacement de son prochain pion en saisissant le numéro de la colonne ciblée. Si la colonne n’existe pas ou si elle est entièrement pleine, le joueur est invité à effectuer un nouveau choix.

Lorsqu’un choix valide est effectué, le tableau est mis à jour et le pion du joueur en cours est placé au sommet de la pile dans la colonne choisie. Une vérification est alors réalisée pour déterminer si le joueur actuel a remporté la partie, si celle-ci est nulle, ou si le jeu se poursuit.

	X							
	1	2	3	4	5	6	7	

FIGURE 5 – Affichage du tableau de jeu du puissance-quatre suite au placement du pion du joueur X dans la première colonne.

Dans tous les cas, le tableau est affiché aux deux joueurs pour inspection. Si un joueur est déclaré gagnant, la configuration gagnante, à savoir un alignement de quatre cases du même joueur, est mise en évidence sur le tableau. Si la partie est nulle, un message adéquat accompagnera le tableau plein. Sinon, le tour passe à l’autre joueur. C’est alors à lui de faire son choix du prochain placement.

	(X)							
	(X)	O						
	(X)	X	O					
	(X)	O	O	O	X			
	1	2	3	4	5	6	7	

FIGURE 6 – Affichage du tableau de jeu du puissance-quatre lorsque le joueur X remporte la partie, avec l’alignement gagnant mis en évidence.

## 3.4 Travail à réaliser

Réaliser l’application de jeu décrite précédemment en respectant les exigences suivantes :

1. Implémenter le moteur de jeu générique pour les jeux à tour de rôle.
2. Développer les deux jeux requis : le tic-tac-toe et le puissance-quatre.
3. Ajouter au moins deux extensions fonctionnelles parmi celles proposées.

### 3.5 Démonstration de l'application souhaitée

Vous pouvez tester deux variantes des jeux demandés. Chaque variante représente une modalité différente et permet de jouer au tic-tac-toe et au puissance-quatre.

Les deux variantes implémentent les extensions de confirmation de placement et de séance de plusieurs parties. La deuxième variante ajoute l'extension de serveur de jeu centralisé.

#### 3.5.1 Application en terminal

Pour des raisons pratiques, cette démonstration utilise une connexion serveur bien qu'elle illustre le mode terminal.

Pour lancer la démonstration, connectez-vous à l'adresse `game-terminal.shahwan.pw` sur le port 8082 en utilisant `netcat` ou un outil équivalent en protocole TCP. La commande suivante utilise `nc` disponible sur la plupart des distributions Linux.

```
nc game-terminal.shahwan.pw 8082
```

Sur Windows, vous pouvez activer `telnet` et l'utiliser de la même manière. L'activation de la commande se fait via les fonctionnalités Windows.

#### 3.5.2 Application avec serveur centralisé de jeu

Pour tester le jeu avec serveur centralisé permettant de jouer contre un adversaire à distance, connectez-vous à l'adresse `game-server.shahwan.pw` sur le port 8082.

```
nc game-server.shahwan.pw 8082
```

## 4 Extensions

Plusieurs extensions vous sont proposées. Deux extensions fonctionnelles au choix sont obligatoires. Les autres sont optionnelles et vous apportent des points bonus.

### 4.1 Extensions fonctionnelles

Les sections suivantes décrivent des fonctionnalités supplémentaires qui peuvent être ajoutées à l'application.

#### 4.1.1 Confirmation de placement

Pour éviter les erreurs de saisie, le joueur est invité à confirmer son choix après chaque saisie valide. La phase de confirmation consiste à afficher un aperçu de l'état du jeu en prenant en compte l'action correspondante, sans pour autant évaluer le résultat de cette action. Ensuite, le joueur est invité à confirmer son choix, par exemple, en appuyant sur Entrée.

Au cours d'une partie de tic-tac-toe, par exemple, lorsqu'un joueur choisit l'emplacement suivant, le tableau s'affiche en indiquant la case ciblée. Aucune vérification n'est faite pour déterminer si la partie est terminée. Le joueur peut confirmer son choix en appuyant sur Entrée. Si le joueur fait une nouvelle saisie, celle-là doit être validée et confirmée à nouveau.

>x<	2	3		
	4	5	6	
	7	8	9	

FIGURE 7 – Exemple de l'affichage du tableau de jeu du tic-tac-toe en phase de confirmation du placement du pion du joueur X dans la case à gauche en haut.

#### 4.1.2 Une séance de plusieurs parties

Les deux joueurs peuvent enchaîner plusieurs parties. Un score est maintenu tout au long de la séance de jeu. Chaque victoire rapporte un point au joueur gagnant, et les parties nulles ne rapportent aucun point. Le score total s'affiche à la fin de chaque partie.

#### 4.1.3 Sauvegarde du jeu

Le jeu peut être sauvegardé à tout moment, par exemple, en saisissant une commande particulière lors d'une action. Au lancement du programme, si une sauvegarde existe, la partie reprend à l'état exact où elle a été sauvegardée.

#### 4.1.4 Partie à distance

Le programme permet une partie ou une séance entre deux joueurs sur des ordinateurs différents. Deux architectures sont possibles :

**Connexion directe** Un joueur lance une partie ou une séance en tant qu'hôte. Le deuxième joueur se connecte en précisant l'adresse réseau de l'hôte.

**Serveur centralisé** Un serveur dédié gère l'orchestration. Les deux joueurs se connectent au serveur pour jouer.

#### 4.1.5 Tic-tac-toe en 3D

Le programme propose le tic-tac-toe 3D comme troisième jeu.

Les règles restent identiques à celles de la version classique. Seul le plateau change : un cube remplace le tableau carré.

Pour afficher ce cube sur un terminal, il est décomposé en 3 tableaux bidimensionnels, chacun représentant un niveau du cube. Les niveaux sont identifiés par les lettres a, b et c.

a1 a2 a3	b1 b2 b3	c1 c2 c3
a4 a5 a6	b4 b5 b6	c4 c5 c6
a7 a8 a9	b7 b8 b9	c7 c8 c9

FIGURE 8 – Affichage du cube de jeu en 3D en configuration initiale.

Le déroulement du jeu est le même que pour la version classique. La seule particularité de la version 3D est le format du libellé : il est composé d'une lettre représentant le niveau, suivie d'un chiffre représentant l'emplacement de la case dans le niveau.

Comme dans la version 2D, un alignement gagnant consiste en 3 cases du même joueur alignées dans n'importe quelle direction : horizontale, verticale ou diagonale, y compris les diagonales traversant les niveaux.

#### 4.1.6 Autres jeux

Le programme propose d'autres jeux, comme les dames et le go.

### 4.2 Extensions techniques

#### 4.2.1 Compilation automatisée

Utilisez un outil de gestion de chaîne de compilation, comme Apache Maven ou Gradle, pour automatiser la compilation, voire l'exécution, de votre programme.

Dans ce cas, vous devez fournir les informations suivantes dans votre rapport.

- Commande à utiliser pour compiler le code et exécuter les éventuels tests unitaires.
- Commande à utiliser pour lancer le programme.

#### 4.2.2 Tests unitaires

Réalisez des tests unitaires qui vérifient le bon fonctionnement des modules principaux de votre programme.