

Projet APO – Tic-tac-toe 3D

Ahmad SHAHWAN

3 janvier 2023

1 Modalités pratiques

Ce TP est à faire en trinôme et est à rendre pour le 3 février 2023.

1.1 Rendu

Vous devrez rendre une archive au format zip contenant les éléments suivants.

- Un rapport au format PDF, de 8 page maximum, qui présentera votre méthodologie, ainsi que les diagrammes UML, les choix de conceptions, etc. Le rapport doit aussi présenter la structure de votre archive.
- Les sources de votre projet dans un dossier à part.
- La Javadoc générée dans un dossier à part.
- Éventuellement, les diagrammes UML individuels dans un dossier à part.

2 Évaluation

Vous serez évalués sur les points suivants :

- Méthodologie (1 point)
- Conception (7 points)
- Code (9 points)
- Documentation (3 points)
- Extensions optionnelles (points en bonus)

2.1 Méthodologie

Vous expliquerez votre méthodologie de travail : articulation entre conception et codage, déroulé temporel du projet, etc. Vous expliquez aussi la répartition des tâches entre les différents membres du groupe.

2.2 Conception

Vous utiliserez les différents types de diagrammes vus en TDs et en cours à bon escient. Votre modèle doit permettre à quelqu'un d'extérieur au projet de le comprendre et de pouvoir le coder. 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 difficulté 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. De plus, les choix majeurs de votre conception devront être justifiés.

2.3 Code

Vous réalisez votre conception en Java. Vous avez la liberté de choisir une des versions LTS suivantes : 8, 11 ou 17. Votre choix de version devra être mentionné explicitement dans votre rapport.

Le code devra être clair : indentation, noms de variables pertinentes, etc. Il devra aussi être efficace autant que possible et en accord avec la conception. Pensez à bien vérifier et traiter les cas d'erreurs dans votre code. Pensez aussi à utiliser les paquetages pour structurer et organiser votre code.

2.4 Documentation

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

2.5 Extensions

Plusieurs extensions optionnelles vous sont proposées. À vous de voir celles qui peuvent vous intéresser. Il est également possible de proposer, et de réaliser, vos propres extensions. Si vous décidez de faire des extensions, votre note maximale potentielle peut dépasser 20. En pratique votre note sera plafonnée à 20 pour le projet, les points supplémentaires seront transférés à votre note de contrôle continu 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

L'objectif de ce projet est de créer un jeu tic-tac-toe, aussi appelé morpion, avec le choix de jouer en 3 dimensions.

3.1 Le jeu

Le tic-tac-toe est un jeu à deux joueurs. À son tour, chaque joueur place une marque l'identifiant, traditionnellement X et O, sur un tableau de 3×3 cellules. Le jeu se termine quand un des deux joueurs gagne ou quand le tableau est tout rempli. Un joueur gagne quand il a placé 3 marques alignées sur le tableau. L'alignement peut être vertical, horizontal ou diagonal.

Dans sa version 3D, le tableau n'est plus représenté par un carré, mais par un cube. Le but du jeu reste l'alignement de 3 cellules. L'alignement peut être parallèle à un axe, diagonal au travers de deux axes, ou diagonal au travers des trois axes.

4 Tronc commun

Vous devez réaliser le jeu tic-tac-toe classique, en 2 dimensions, ainsi qu'une extension du même jeu en 3 dimensions.

Tout au début, le programme demande à l'utilisateur de choisir la version souhaitée du jeu : en 2D, ou en 3D. Les deux joueurs alternent ensuite à utiliser l'entrée standard pour jouer leurs tours respectifs. L'affichage du jeu s'effectue entièrement sur la sortie standard.

4.1 Jeu 2D

Une matrice représentant le tableau s'affiche à l'écran. Pour chaque cellule inoccupée un libellé variant de 1 à 9 est affiché à l'intérieur de la cellule. Le libellé 1 correspond à la cellule dans le coin en haut à gauche, le libellé 9 correspond à la cellule dans le coin diagonalement opposé. Une cellule occupée, c'est-à-dire déjà marquée par un des deux joueurs, contient le libellé de celui-ci. Les deux joueurs sont respectivement identifiés par les libellés X et O.

```
| 1  2  3 |  
| 4  5  6 |  
| 7  8  9 |
```

FIGURE 1 – Affichage du tableau de jeu en 2D en configuration initiale.

À son tour, il est demandé au joueur de choisir son prochain placement en entrant le libellé de la cellule ciblée. Si le libellé est inexistant, ou si la cellule est occupée, il est demandé à nouveau au joueur de faire son choix.

Une fois que le joueur a fait un choix valide, il lui sera demandé de confirmer son choix. Pour ce faire, la matrice est affichée à nouveau avec la cellule choisie identifiée visuellement (par exemple par des guillemets, flèches, ou points d'interrogation). Le joueur peut ensuite confirmer son choix en appuyant sur la touche Entrer. Il peut aussi faire un nouveau choix en donnant un nouveau libellé, dans ce cas, ce choix sera à confirmer à nouveau.

```
|>x< 2  3 |  
| 4  5  6 |  
| 7  8  9 |
```

FIGURE 2 – Exemple de l'affichage du tableau de jeu en 2D en configuration de confirmation du placement de la marque du joueur X dans la cellule à gauche en haut.

Lorsque le choix est confirmé, la matrice sera mise à jour avec la cellule choisie ayant la marque du joueur en cours. À ce moment-là une vérification est effectuée pour savoir si le joueur actuel a gagné la partie, ou si la partie est finie car le tableau est rempli.

Dans l'un de ces deux cas un message adéquat est affiché, et le programme est terminé. Sinon, le tour passe à l'autre joueur. C'est à lui alors de faire son choix du prochain placement.

	X	2	3	
	4	5	6	
	7	8	9	

FIGURE 3 – Affichage du tableau de jeu en 2D suite à la confirmation du placement de la marque dans la cellule à gauche en haut par le joueur X.

4.2 Jeu 3D

La version 3D du jeu garde les mêmes principes que ceux de la version classique. Sa spécificité est le fait que le tableau, traditionnellement carré, est représenté en cube.

Pour permettre d'afficher un tel objet 3D sur la sortie standard, le cube est éclaté en 3 matrices deux-dimensionnelles. Chaque matrice représentant un niveau du cube. Les niveaux sont identifiés par des lettres a, b, et c.

(a)	(b)	(c)
1 2 3	1 2 3	1 2 3
4 5 6	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9

FIGURE 4 – Affichage du cube de jeu en 3D en configuration initiale.

La suite du jeu est quasi-identique pour les deux versions. Seule particularité pour la version 3D est le fait que le libellé de la cellule choisie est composé d'une lettre représentant le niveau suivi par un chiffre représentant l'emplacement de la cellule dans le niveau. Par exemple, si le joueur fourni b3 comme libellé, c'est la cellule en haut à droite du niveau b qui sera choisie.

(a)	(b)	(c)
[X] 2 3	[X] 0 0	[X] 2 3
4 5 6	4 5 6	4 5 6
7 8 9	7 8 9	7 8 9

Alignement parallèle à un axe.

(a)	(b)	(c)
1 [X] 3	1 0 0	1 2 3
4 5 6	4 [X] 6	4 5 6
7 8 9	7 8 9	7 [X] 9

Alignement diagonal au travers de deux axes.

(a)	(b)	(c)
[X] 2 3	1 0 0	1 2 3
4 5 6	4 [X] 6	4 5 6
7 8 9	7 8 9	7 8 [X]

Alignement diagonal au travers des trois axes.

FIGURE 5 – Exemples des configurations gagnantes pour joueur X. Les cellules des combinaisons gagnantes sont entourées par des crochets.

4.3 Travail à Réaliser

Réaliser le jeu décrit auparavant. Quelques conseils peuvent vous être utiles :

- Commencez par réaliser le jeu en 2D. Une fois que vous êtes confiants de votre code, vous pouvez le généraliser pour le cas en 3D.
- Pour la factorisation du code, l'héritage et la réalisation des interfaces seront sûrement pertinents. Cependant, l'héritage seul ne suffira pas pour factoriser certains comportements. Pensez à coupler l'héritage avec les types génériques pour accomplir une factorisation optimale.

5 Extensions optionnelles

5.1 Extensions fonctionnelles

5.1.1 Taille variée

1 point

Faites en sorte que la taille du tableau de jeu soit paramétrable. Par défaut, la taille d'un tableau classique est 3^2 et d'un tableau 3D est 3^3 . Un tableau d'une taille variée acceptera un paramètre n . La taille d'un tableau

classique sera donc n^2 , et d'un tableau 3D n^3 .

5.1.2 Affichage de la combinaison gagnante

1 point

Lorsqu'un joueur gagne, et en plus d'afficher un message de félicitation, le programme doit aussi afficher le tableau de jeu en montrant la combinaison gagnante. Par exemple, vous pouvez entourer les cellules alignées par des parenthèses ou des crochets.

5.1.3 Sauvegarde du jeu

1 point

Le jeu peut être sauvegardé à tout moment, par exemple en fournissant une commande particulière au lieu du libellé d'une cellule.

Quand le programme se lance, et si un jeu a été sauvegardé, le jeu reprend de là où il était arrêté. Libre à vous de choisir le format de sauvegarde.

5.1.4 Intelligence artificielle

2 points

En plus du mode humain contre humain, le programme proposera le mode humain contre ordinateur. Afin de faire une IA capable de jouer, vous pouvez regarder les algorithmes minimax, ou encore MCTS (Monte Carlo Tree Search).

5.1.5 Partie à distance

2 points

Le programme proposera 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.

5.2 Extensions techniques

5.2.1 Compilation automatisée

1 point

Utilisez un outil de gestion de chaîne de compilation, comme Apache Maven ou Gradle, pour automatiser la compilation, voire le lancement, 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.

5.2.2 Versionnement du code

1 point

Utilisez l'outil de versionnement git pour permettre de travailler en groupe et de garder l'historique de votre travail. Si le répertoire git utilisé est public et accessible, son adresse doit être fournie dans le rapport. Sinon, assurez-vous d'avoir inclus le dossier `.git` dans votre livrable.

5.2.3 Tests unitaires

1 point

Réalisez des tests unitaires qui vérifient la bonne exécution des méthodes principales de votre programme.