

## Introduction

Le jeu de tic-tac-toe, aussi nommé « morpion » en France et « oxo » en Belgique, est bien connu. Ce jeu de réflexion à deux joueurs se joue sur un plateau de 3x3 et le but consiste à aligner 3 pièces du même joueur de façon horizontale, verticale ou diagonale. Il est relativement simple pour un ordinateur de jouer parfaitement et ainsi gagner ou annuler à toutes les parties. Dans le cadre de ce laboratoire, nous allons utiliser une version plus complexe dans laquelle chaque case du tic-tac-toe est un autre plateau de tic-tac-toe. Il s'agit donc de deux plateaux imbriqués. Ceci rend l'arbre de recherche plus complexe et une stratégie plus complexe, incluant une vue globale du jeu, doit être définie.

Les objectifs de ce laboratoire sont :

- Implémentation de l'algorithme minimax et alpha-beta
- Implémentation d'heuristiques pour le jeu de Tic tac toe géant
- Exploration et implémentation de différentes stratégies afin d'améliorer l'efficacité de votre algorithme de décision.

## Règles du jeu

### Configuration du plateau

Le tic-tac-toe géant est un plateau composé de 9 petits plateaux de tic-tac-toe positionnés dans une grille de 3x3. Les joueurs jouent leurs coups sur les petits jusqu'à ce qu'un des deux joueurs gagne la partie. Au départ, le plateau est complètement vide comme le montre la figure 1. Noter que les « X » pâles sont les coups possibles pour le joueur « X » qui au départ peut jouer n'importe où sur le plateau.

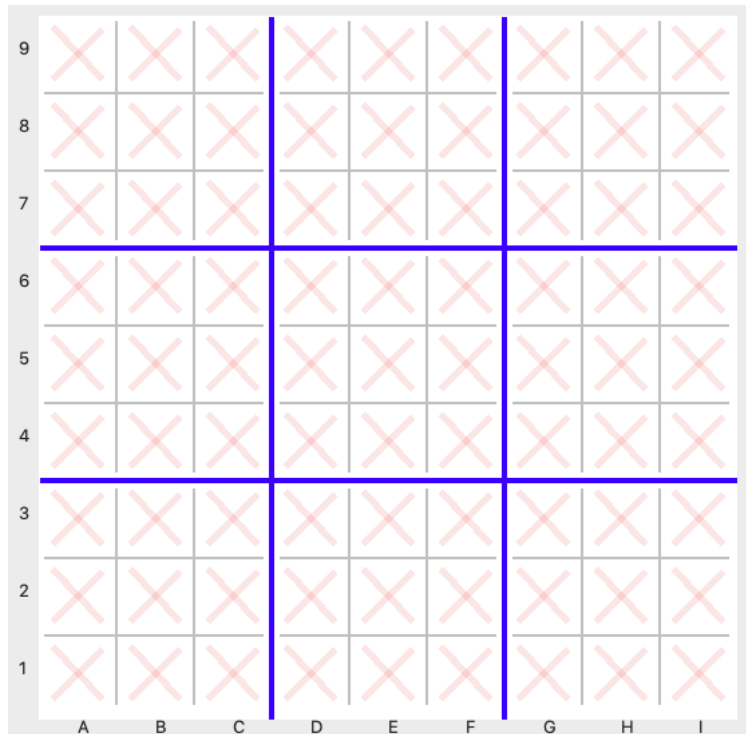


Figure 1 : Configuration initiale du jeu de « tic-tac-toe géant »

## Règles du jeu

Les petits tic-tac-toe sont appelés plateaux locaux et le grand tic-tac-toe est appelé plateau global. Le but du jeu est d'obtenir un tic-tac-toe dans le plateau global en alignant des victoires dans les plateaux locaux. La figure 2 montre un exemple de plateau gagnant pour le joueur « X ».

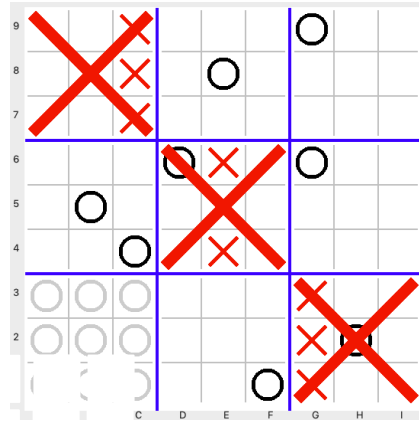


FIGURE 2 . Exemple de configuration gagnante pour les « X »

Le placement des pièces est plus compliqué que dans le tic-tac-toe normal. Le joueur « X » démarre toujours la partie et peut jouer sur n'importe laquelle des 81 cases du plateau de jeu. Ce mouvement détermine l'emplacement dans lequel l'opposant pourra jouer. En effet, la position du coup fait par un joueur sur le plateau local détermine le plateau local sur lequel l'opposant pourra jouer. Par exemple, si le joueur « X » joue dans le coin supérieur gauche de son plateau local, l'opposant devra jouer sur le plateau local supérieur gauche du plateau global. La figure 3 montre cet exemple.

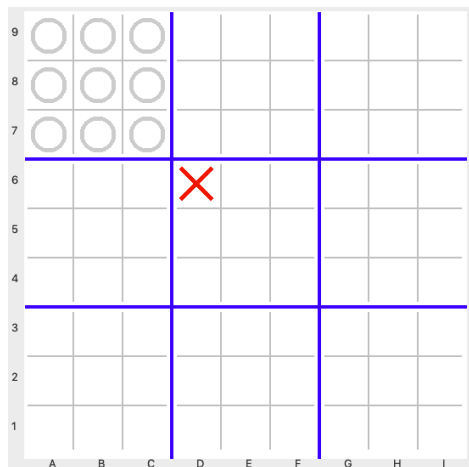


Figure 3 : Le joueur « X » a joué le coin supérieur gauche du plateau local central, l'adversaire devra donc jouer sur le plateau supérieur gauche du plateau global.

Par contre, lorsqu'un plateau local est gagné par un joueur (ou qu'il n'y a plus de position libre), ce plateau devient fermé (ne peut plus être utilisé). Donc, si un joueur joue un coup qui mène vers un plateau fermé, le joueur adverse pourra jouer dans n'importe quelle case vide de n'importe quel plateau local encore ouvert. La figure 4 donne un exemple de ce cas de figure.

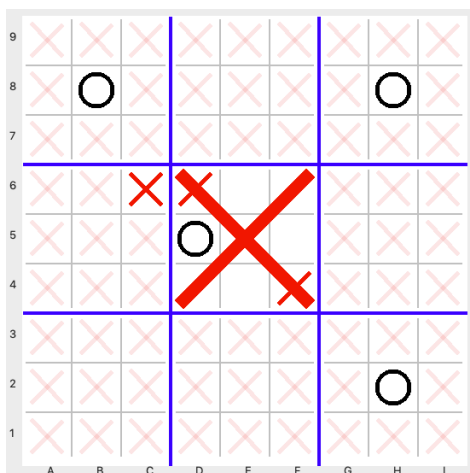


Figure 4 : La case centrale du plateau global est fermée puisqu'elle a été gagnée par le joueur « X ». Par contre, en jouant en H2, le joueur « O » force l'adversaire à jouer sur un plateau fermé. En conséquence, le joueur « X » peut jouer sur n'importe quelle case du plateau global encore ouverte.

Il est facile de voir que cette dernière option peut donner un avantage certain à l'adversaire. Il est donc nécessaire de bien analyser les coups joués localement parce qu'ils ont une grande influence sur le plateau global.

### Autres règles et consignes

Votre programme devra vérifier la validité du coup entrée par l'adversaire. De plus, votre programme aura un maximum de 3 secondes à chaque tour pour jouer son coup. Votre programme devra offrir la possibilité au joueur de jouer les « X » ou les « O » (en option sur la ligne de commande ou en posant la question au début de la partie).

Votre programme devra aussi permettre de spécifier l'adresse IP du serveur.

## Interface

Une interface graphique de type serveur est disponible sur le site du cours. Il est cependant recommandé d'avoir une interface console minimale afin de pouvoir suivre le déroulement du jeu sur la console. Cette interface pourrait être pratique pour le "debugage" de votre application.

La communication avec le serveur est plutôt simple. Au départ, le serveur vous envoie le message "1" avec la configuration du plateau de départ si vous êtes le joueur « X ». Si par contre vous êtes le joueur « O » la commande reçue sera le "2" suivit de la configuration du plateau. Dans ce cas-ci, la configuration initiale est une série de 0 étant donné que le plateau initial est vide.

Les différentes pièces sont représentées par des chiffres :

Numéro	Pièce
0	Case vide
2	« O »
4	« X »

Lorsque c'est le tour à votre programme de jouer son coup, le serveur vous envoie le message "3", ainsi que le dernier coup jouer par l'adversaire. Voici un exemple de message :

3 D6

Si vous êtes le joueur « X », vous devez donc jouer le premier. Dans ce cas particulier, le message va contenir un coup invalide :

3 A0

Dans l'éventualité où le coup transmis serait invalide (cela ne devrait jamais se produire), le message "4" sera envoyé par le serveur qui restera en attente d'un nouveau coup.

### Spécification d'un coup

Dans le jeu de tic-tac-toe géant, un coup est simplement l'identification de la case sur laquelle le pion est ajouté. Il suffit donc de transmettre la case désirée au serveur.

Un programme de démonstration, en Java, est disponible sur le site du cours.

## Fonctions à implémenter

Voici les fonctions qui doivent minimalement être implémentées afin d'avoir un joueur artificiel :

### Générateur de mouvements :

Cette fonction utilise la configuration du plateau pour générer tous les coups possibles. Tous les coups de cette liste doivent être valides.

### Fonction d'évaluation :

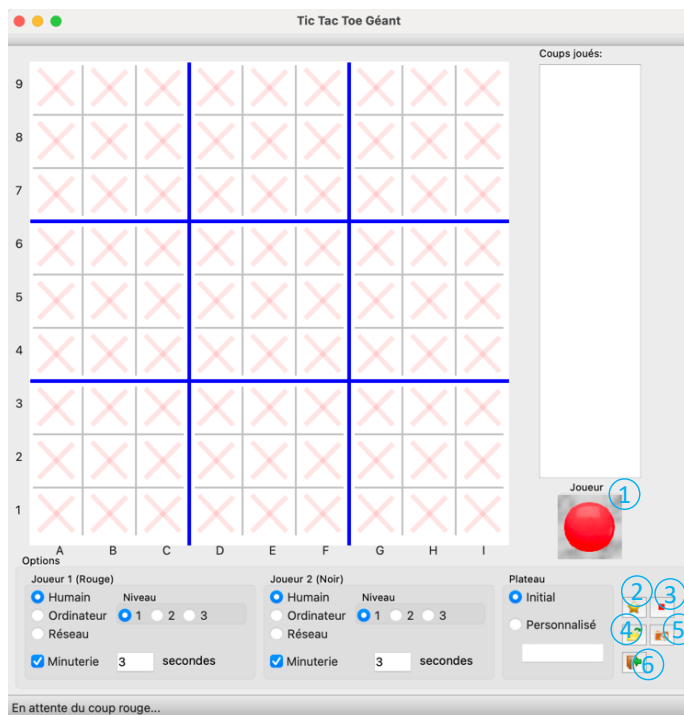
Cette fonction a pour but d'évaluer la position sur le plateau. Plus l'évaluation est bonne, plus votre programme sera bon. C'est aussi cette fonction qui détermine si un joueur a gagné.

### Algorithme MinMax et AlphaBeta :

Fonction qui détermine le prochain coup de l'ordinateur. Je vous rappelle que cette fonction dispose de 3 secondes pour choisir son coup.

## Serveur

La figure ci-dessous montre l'interface graphique du serveur. Les principales fonctionnalités y sont décrites.



- ① Indique qui doit jouer le prochaine coup
- ② Démarre une nouvelle partie. Si l'option « plateau personnalisé » est activé, ce sera ce plateau qui sera utilisé comme point de départ. Attention, ceci est pour débbugger, l'IA du serveur pourrait ne pas fonctionner correctement avec un plateau personnalisé.
- ③ Permet d'arrêter la partie en cours. Cette option n'est pas encore vraiment opérationnelle.
- ④ Pour charger un plateau personnalisé
- ⑤ Pour sauvegarder un plateau personnalisé
- ⑥ Quitter l'application

Au départ du serveur, le jeu est mode humain contre humain. Pour démarrer une nouvelle partie, il faut d'abord modifier les types de joueurs (humain, ordinateur ou réseau) et les niveaux de jeux. Ensuite, le bouton étoile (le 2 sur l'image), permet de démarrer une nouvelle partie avec cette configuration.

Lors d'une partie réseau, le serveur attend la connexion du client. S'il s'agit d'une partie entre deux joueurs de type réseau, la première connexion sera pour le joueur rouge et la deuxième pour le joueur noir.

## Vérification du programme

---

Une première vérification sera effectuée à la 3<sup>e</sup> semaine. À ce moment, vous devrez démontrer le fonctionnement de votre programme. À ce point, vous devez avoir implémenté :

- L'algorithme minmax
- L'élagage alpha-beta
- Une fonction d'évaluation statique de base

À la dernière séance de laboratoire, un tournoi est organisé. Votre programme sera évalué en fonction de votre performance dans le tournoi, mais aussi en fonction des performances contre le programme de démonstration.

Noter que pour compléter la grille du tournoi, le programme de démonstration, au niveau 2, peut-être ajouté à la compétition.

## Rapport de laboratoire

---

Les modalités du rapport de laboratoire vous seront spécifiées par le chargé de laboratoire vers la fin de la session.

## Barème de correction

---

Le barème de correction est le suivant :

Tic tac toe	3
Point de contrôle 1	1
Point de contrôle 2 (niveau 1)	2
Battre le serveur au niveau 2	2
Battre le serveur au niveau 3 (bonus)	(2)
Performance dans le tournoi	4
Rapport de laboratoire	3
Total	15

Pour obtenir tous vos points dans les évaluations contre le serveur, vous devez le battre en tant que joueur « X » et joueur « O ». Vous avez droit à deux essais par couleur.

Noter que les points bonus ne permettent pas de dépasser le 15/15.

Le barème concernant les points de performance sera expliqué par le responsable du laboratoire.