

# TP noté : Le jeu du Moulin

L'objectif de ce TP est de programmer le jeu du moulin entre joueurs humains ou contre l'ordinateur.  
Ce TP est à programmer en binôme.

## Présentation du jeu

Le jeu du moulin existe depuis l'Egypte Antique.

Chaque joueur dispose de 9 pions de sa couleur. L'objectif du jeu est de retirer les pions de l'adversaire sur le damier ou de le bloquer afin qu'il ne puisse plus jouer.

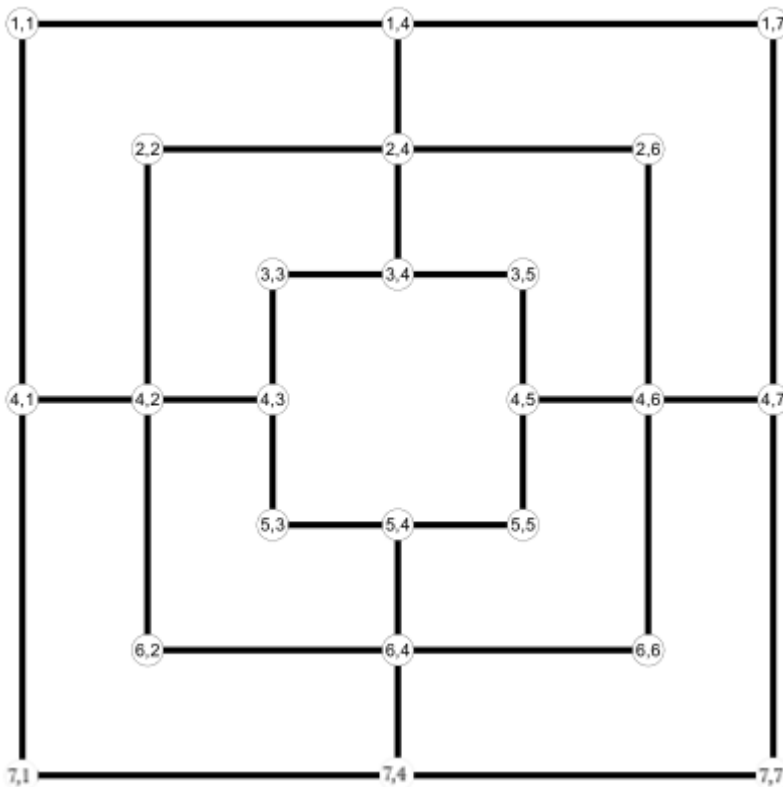
Les règles du jeu sont disponibles sur cette page :

[https://fr.wikipedia.org/wiki/Jeu\\_du\\_moulin#R%C3%A8gles\\_du\\_jeu](https://fr.wikipedia.org/wiki/Jeu_du_moulin#R%C3%A8gles_du_jeu)

## Partie 1 : Moteur de jeu

### Considérations importantes

- Dans un premier temps, on se limitera à une application en mode console (sans interface graphique).
- On limitera également le jeu à la phase 1 (pose des pions) et la phase 2 (mouvement) : le saut ne fera pas partie des actions possibles. (voir règles du jeu)
- La position des pions sur le damier sera représentée par des coordonnées (X,Y). Ces coordonnées doivent suivre le schéma ci-dessous. Par exemple, un pion placé en bas à droite est identifié par les coordonnées (7,7)



Créer un modèle objet pour représenter l'état du damier à chaque instant. Attention à bien respecter les coordonnées imposées sur le damier (voir considérations importantes) !

## Moteur de jeu

Développer le moteur de jeu.

Ce moteur doit permettre au joueur courant de réaliser une des deux actions : pose ou mouvement (si autorisé). Suite à cette action, le modèle du plateau est mis à jour et c'est au 2e joueur de jouer.

Le moteur de jeu doit également calculer si la partie est terminée ou non et déclarer le vainqueur.

**Attention : la cohérence du programme repose essentiellement sur le bon fonctionnement du moteur ! Ne pas négliger les tests dans cette partie**

## Sauvegarde / Restauration du plateau

En utilisant la sérialisation et désérialisation d'objets JSON, coder un système de sauvegarde et de restauration du plateau de jeu.

## Intelligence artificielle

Développer une intelligence artificielle pour le jeu, permettant de jouer seul contre l'ordinateur.

L'interface de cette IA doit être un module `ia` respectant le contrat suivant :

- `ia.new_game(ia_first: bool)` informe l'IA qu'une nouvelle partie démarre. Le paramètre `ia_first` vaut `True` si l'IA est le premier joueur, et `False` si le joueur humain commence.
- `ia.player_sets(x, y)` informe l'IA que le joueur humain pose un pion en position (X,Y).
- `ia.player_moves(x1, y1, x2, y2)` informe l'IA que le joueur humain déplace un pion de la position (X1,Y1) à la position (X2,Y2). Si la phase de saut est développée, c'est également cet appel qui est utilisé.
- `ia.play()` est la fonction demandant à l'ordinateur quel est son prochain coup. La réponse peut être de deux types :
  - `"set", (x, y)` demande à placer un pion sur la case (X,Y).
  - `"move", (x1, y1), (x2, y2)` demande à déplacer un pion de la case (X1,Y1) à la case (X2,Y2).

Dans un premier temps, on pourra commencer par chercher l'ensemble des coups possibles. Une première version de l'IA pourra choisir un coup au hasard, avant d'être améliorée.

## Ajout du saut

Améliorer le moteur de jeu et l'IA pour prendre en compte l'étape 3 du jeu : le saut (voir règles).

## Partie 2 : Interface graphique

Développer une interface graphique pour le jeu.

Cette interface aura les contraintes suivantes :

- utiliser la librairie Tkinter.

- afficher :
  - le plateau de jeu (le damier)
  - les différents pions sur le damier
  - le gagnant en cas de victoire
- permettre de charger et sauvegarder une partie

Le joueur courant doit également pouvoir placer un pion ou le déplacer en utilisant l'interface graphique.

Le développement de l'interface graphique peut être intégré dès la création du moteur de jeu.