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

Clarification des règles

Le jeu est en deux phases.

1. Pour commencer, les joueurs jouent à tour de rôle une pièce de leur propre couleur sur n'importe quel point inoccupé jusqu'à ce que les dix-huit pièces aient été jouées (9 pions par joueur).
2. Après cela, le jeu continue alternativement mais chaque tour consiste en un joueur qui déplace une pièce le long d'une ligne (horizontale ou verticale) vers un point adjacent (sauf s'il lui reste 3 pièces, voir plus loin).

Au cours de ces deux phases, chaque fois qu'un joueur atteint un moulin (3 pions alignés horizontalement ou verticalement), ce joueur retire immédiatement du plateau une pièce appartenant à l'adversaire et **qui ne fait pas partie d'un moulin**. Si toutes les pièces adverses forment des moulins, une exception est faite et le joueur est autorisé à retirer n'importe quelle pièce.

Ce n'est que lors de la formation d'un moulin qu'une pièce est capturée mais un joueur brise souvent un moulin en en déplaçant une pièce puis, dans un tour suivant, la joue de nouveau, formant ainsi un nouveau moulin et capturant une autre pièce.

Les pièces capturées ne sont jamais rejouées sur le plateau et restent capturées pour le reste de la partie.

Règle du saut : lorsqu'un joueur est réduit à seulement trois pièces, il est autorisé à se déplacer de n'importe quel point vers n'importe quel autre point du plateau, quelles que soient les lignes ou si le point de destination est adjacent ou non.

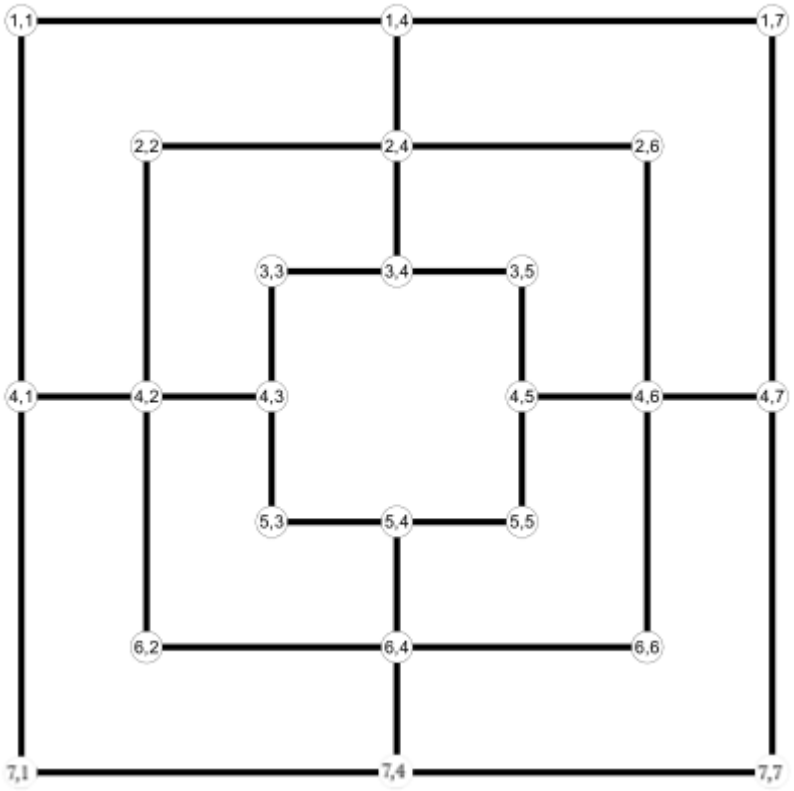
Le jeu est terminé lorsqu'un joueur perd en étant réduit à deux pièces ou en étant incapable de bouger.

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)



Modèle

> 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,u=None,v=None)` informe l'IA que le joueur humain pose un pion en position (X,Y). Les paramètres (U,V) correspondent à la position du pion appartenant à l'IA et volé par le joueur humain, dans le cas où un pion a été volé pendant cette phase (sinon ces paramètres valent None).
- `ia.player_moves(x1,y1,x2,y2,u=None,v=None)` 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. Les paramètres (U,V) correspondent à la position du pion appartenant à l'IA et volé par le joueur humain, dans le cas où un pion a été volé pendant cette phase (sinon ces paramètres valent None).
- `ia.play()` est la fonction demandant à l'ordinateur quel est son prochain coup. La réponse peut être de deux types :
  - `"set",(x,y),(u,v)` demande à placer un pion sur la case (X,Y) et à voler le pion en position (u,v). Si aucun pion n'a été volé, alors (u,v) vaut (None,None).
  - `"move",(x1,y1),(x2,y2),(u,v)` demande à déplacer un pion de la case (X1,Y1) à la case (X2,Y2) et à voler le pion en position (u,v). Si aucun pion n'a été volé, alors (u,v) vaut (None,None).
  - Attention : les retours de ces méthodes sont des tuples (et non une simple chaîne de caractères.Par exemple `"set",(x,y),(u,v)` est de type `tuple(str, tuple(int,int), tuple(int,int))`

Exemples d'appels effectués par le joueur humain (exemples non consécutifs, on suppose le tableau dans un état permettant ces appels) :

```
ia.player_sets(2,4) # place un pion en position (2,4)
ia.player_sets(2,4,2,2) # place un pion en position (2,4). Ce pion forme un
moulin et permet de voler un pion de l'IA en position (2,2). L'IA pourra
vérifier la validité de cet appel.
ia.player_moves(2,4,2,2) # déplace le pion (2,4) en position (2,2).
ia.player_moves(2,4,2,2,3,4) # déplace le pion (2,4) en position (2,2) et
vole un pion de l'IA en position (3,4).
```

Exemples de retours de l'IA :

```
ia.play() -> "set",(2,4),(None,None) # L'IA place un pion en position
(2,4).
ia.play() -> "set",(2,4),(2,2) # L'IA place un pion en position (2,4) et
vole un pion du joueur humain en position (2,2).
ia.play() -> "move",(2,4),(2,2) # L'IA déplace un pion de la position (2,4)
vers la position (2,2).
ia.play() -> "move",(2,4),(2,2),(3,4) # L'IA déplace un pion de la position
(2,4) vers la position (2,2) et vole un pion du joueur humain en position
(3,4).
```

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 régles).

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.