RAPPORT PROJET DIA : MORPION Ultime

VAIANI-CANON Alexandre, THOMINE Bérénice, THOMASSIN Pablo A3S6 Groupe S

Dans la suite de ce rapport nous allons détailler l'utilité de chaque méthode et classe implémenté au sein du programme, nous commencerons par évoquer les difficultés rencontrées et surtout les objectifs du projet.

Pour le programme, l'implémentation la plus compliqué fut celle de donner une action valide jouer par le minmax, en effet nous avions un programme de morpion 3X3 fonctionnel avec un minmax performant, mais pour le transiter vers une grille 9X9 nous avons eu de nombreuse erreur de gestion d'indice pour lui faire jouer des coups corrects.

Nous avons donc, décidé de faire les choix suivants pour le développement du programme :

Classe TicTacToe : Permettant de faire la gestion des sous plateaux

Méthode __init__(self)

- Cette méthode est le constructeur de la classe TicTacToe.
- Elle initialise la grille du jeu avec une liste de listes de chaînes de caractères, représentant un tableau de jeu vide.

Méthode coup(self, ligne, colonne, symbole)

- Cette méthode permet de placer un coup sur la grille du jeu.
- Elle prend en paramètres les coordonnées de la case où le coup doit être placé (ligne et colonne) ainsi que le symbole du joueur qui joue.
- Si la case est vide, le coup est placé sur la grille et la méthode retourne True. Sinon, la méthode retourne False.

Méthode victoire(self, symbole)

- Cette méthode vérifie si un joueur a remporté la partie en vérifiant les différentes combinaisons gagnantes.
- Elle prend en paramètre le symbole du joueur à vérifier.
- La méthode parcourt la grille et vérifie si le symbole du joueur est présent sur une ligne, une colonne ou une diagonale.
- Si une combinaison gagnante est trouvée, la méthode retourne True. Sinon, elle retourne False.

Méthode plein(self)

- Cette méthode vérifie si la grille du jeu est pleine, c'est-à-dire si toutes les cases sont remplies.
- Elle parcourt la grille et vérifie si une case vide est présente.

- Si une case vide est trouvée, la méthode retourne False. Sinon, elle retourne True.

Classe UltimateTicTacToe : Pour pouvoir jouer sur une grille 9X9

Méthode __init__(self)

- Cette méthode est le constructeur de la classe UltimateTicTacToe.
- Elle initialise le jeu en créant une grille de neuf jeux TicTacToe disposés en une grille 3x3.
- Elle initialise également d'autres attributs tels que le symbole actuel, la grille actuelle, le compteur de coups et la grille des gagnants.

Méthode coup(self, grille_ligne, grille_colonne, ligne, colonne)

- Cette méthode permet de placer un coup dans le jeu UltimateTicTacToe.
- Elle prend en paramètres les coordonnées de la grille et de la case où le coup doit être placé, ainsi que le symbole du joueur qui joue.
- La méthode vérifie si le coup est valide en fonction des conditions suivantes :
    - Le compteur de coups doit être supérieur à zéro (sauf pour le premier coup).
    - La grille actuelle doit correspondre à la grille où le coup est joué (sauf pour le premier coup).
    - La case de la grille choisie ne doit pas être déjà gagnée ou pleine.
- Si le coup est valide, il est placé sur la grille correspondante et la méthode met à jour les attributs pertinents (grille actuelle, symbole actuel, compteur de coups, etc.).
- Si le coup conduit à une victoire dans la grille choisie, le gagnant de cette grille est enregistré.
- La méthode retourne True si le coup est placé avec succès, sinon elle retourne False.

Méthode victoire(self)

- Cette méthode vérifie si un joueur a remporté la partie dans l'ensemble du jeu UltimateTicTacToe.
- Elle parcourt les grilles gagnantes dans toutes les directions et vérifie si un joueur a remporté une ligne, une colonne ou une diagonale.
- Si un joueur a remporté la partie, la méthode retourne le symbole du joueur gagnant. Sinon, elle retourne None.

Méthode plein(self)

- Cette méthode vérifie si toutes les grilles du jeu UltimateTicTacToe sont pleines.
- Elle parcourt toutes les grilles et vérifie si une grille n'est pas pleine.
- Si une grille n'est pas pleine, la méthode retourne False. Sinon, elle retourne True.

Méthode afficher(self)

- Cette méthode affiche l'état actuel du jeu UltimateTicTacToe.
- Elle parcourt les grilles et les cases du jeu et les affiche à l'écran.

Méthode actions(self)

- Cette méthode retourne la liste de toutes les actions possibles dans le jeu UltimateTicTacToe.
- Les actions possibles sont représentées par des tuples contenant les coordonnées des grilles et des cases où un coup peut être joué.

Méthode result(self, action)

- Cette méthode retourne un nouvel état du jeu UltimateTicTacToe résultant d'une action donnée.
- Elle crée une copie profonde du jeu actuel, effectue l'action donnée et retourne le nouvel état.

Méthode terminal_test(self)

- Cette méthode vérifie si le jeu UltimateTicTacToe a atteint un état terminal, c'est-à-dire si un joueur a gagné ou si le jeu est plein.
- Si le jeu est terminé, la méthode retourne True. Sinon, elle retourne False.

Méthode utility(self)

- Cette méthode retourne une valeur d'utilité pour l'état actuel du jeu UltimateTicTacToe.
- Si un joueur a gagné, la méthode attribue une valeur élevée au joueur X et une valeur basse au joueur O.
- Sinon, la méthode attribue un score basé sur les positions des symboles dans les grilles. Les positions centrales et les alignements de symboles sont valorisés.
- La méthode retourne le score d'utilité calculé.

Méthode minimax_alphabeta(self, depth, player, alpha, beta) : L'implémentation de l'ia

- Cette méthode implémente l'algorithme du minimax avec élagage alpha-bêta pour trouver le meilleur coup à jouer.
- Elle utilise la récursivité pour explorer les différents états du jeu et attribuer des valeurs d'utilité.
- La méthode prend en compte le joueur actuel, la profondeur de recherche et les valeurs alpha et bêta pour effectuer l'élagage.
- Elle retourne le meilleur coup possible et sa valeur d'utilité.

Méthode jouer_IA(self, depth)

- Cette méthode permet à l'IA de jouer un coup dans le jeu UltimateTicTacToe.
- Elle utilise l'algorithme minimax avec élagage alpha-bêta pour choisir le meilleur coup en fonction de la profondeur spécifiée.
- Le coup choisi est joué dans le jeu.

Méthode choisir_premier_joueur(self)

- Cette méthode permet à l'utilisateur de choisir le premier joueur, soit l'humain soit l'IA.
- Elle demande à l'utilisateur de saisir "H" pour l'humain ou "I" pour l'IA.
- En fonction du choix de l'utilisateur, la méthode retourne le symbole correspondant au joueur humain ("X" dans le code).

Fonction jouer()

- Cette fonction principale permet de jouer au jeu UltimateTicTacToe.
- Elle crée une instance de UltimateTicTacToe, demande à l'utilisateur de choisir le premier joueur et boucle jusqu'à ce qu'un joueur gagne ou que le jeu soit plein.

- À chaque tour, l'état du jeu est affiché et le joueur correspondant doit effectuer un coup.
- Si c'est le tour de l'humain, l'entrée de l'utilisateur est demandée pour choisir le coup.
- Si c'est le tour de l'IA, l'IA utilise l'algorithme minimax avec élagage alpha-bêta pour choisir le meilleur coup.
- Une fois la partie terminée, le résultat est affiché.