

MORIARTY.PY

I) FONCTIONS DE BASE

I.1) CREATE_INITIAL_GRID()

Crée la grille de jeu (8x8) en initialisant les positions :

- (3,3) → [I]
- (4,4) → [I]
- (4,3) → [A]
- (3,4) → [A]

Args :

- None

Return :

- grid → new grid

I.2) CREATE_PLAYER()

Crée les nouveaux joueurs et leur attribue un rôle (Activator / Inhibitor) avec des noms par défaut s'ils ne sont pas renseignés.

Args :

- None

Return :

- activator → nom du joueur Activator
- inhibitor → nom du joueur Inhibitor

I.3) DISPLAY_GRID(GRID)

Affiche la grille actuelle.

Args :

- grid → grille actuelle

Return :

- None → modifie directement la grille depuis la fonction

I.4) CHECK_END(CASE, NO_CASE, GRID)

Vérifie que la partie peut continuer en testant toutes les positions possibles pour le joueur actuel.

Args :

- case → valeur de la case du joueur actuel
- no_case → valeur de la case du joueur ne jouant pas actuellement
- grid → grille actuelle

Return :

- True → si un mouvement permet de continuer la partie
- False → si aucun mouvement ne permet de continuer la partie

I.5) CHECK_CHANGES(COORDXY, GRID, CASE = NONE, NO_CASE = NONE)

Vérifie les changements induits par les coordonnées testées.

Args :

- coordXY → coordonnées à tester
- grid → grille actuelle
- case → valeur de la case du joueur actuel
- no_case → valeur de la case du joueur ne jouant pas actuellement

Return :

- status → renvoie True si le mouvement induit un changement sinon renvoie False
- taken_cases → renvoie les coordonnées des cases acquises par le mouvement actuel

I.6) UPDATE_GRID(GRID, TAKEN_CASES, CASE)

Met à jour la grille suite au dernier mouvement.

Args :

- grid → grille actuelle avant les changements
- taken_cases → coordonnées des points subissant une modification après le mouvement
- case → valeur de la case du joueur actuel

Return :

- None → modifie directement la grille actuelle avec les différentes positions acquises

I.7) GET_ALLOWED_POSITIONS(COORDXY, GRID)

Calcule les différents « anneaux » (+1, +2) autour du point de coordonnées passé en paramètres.

Args :

- coordXY → coordonnées entrée par l'utilisateur
- grid → grille actuelle

Return :

- testing_coord → renvoie les coordonnées de l'anneau +2
- surrounding_coord → renvoie les coordonnées de l'anneau +1

I.8) GET_POSITION(C_PLAYER, CASE, NO_CASE, GRID)

Récupère la position entrée par le joueur actuel et vérifie que celle-ci est valide.

Args :

- c_player → nom du joueur actuelle
- case → valeur de la case du joueur actuel
- no_case → valeur de la case du joueur ne jouant pas actuellement
- grid → grille actuelle

Return :

- (CoordX, CoordY) → coordonnées validées pour un mouvement
- grid → grille actuelle
- taken_cases → cases prises suite au mouvement

I.9) PLAYING(PPLAYER, GRID)

Lance un tour de jeu.

Args :

- player → joueur entrant dans son tour de jeu
- grid → grille actuelle

Return :

- grid → grille mise à jour
- end → True si la partie est terminée, False sinon

II) INTELLIGENCE ARTIFICIELLE

II.1) EASY_MODE()

Intelligence artificielle basique ne favorisant pas les mouvements rapportant le plus de points.

II.2) HARDCORE_MODE()

Intelligence artificielle plus évoluée favorisant les mouvements rapportant le plus de points.