

Documentation des prédicats

Description de la modélisation :

Le plateau de Othello est vue comme une liste de 64 éléments ou l'élément est soit :

1. **Noir**
2. **Blanc**
3. **Libre**

01	02	03	04	05	06	07	08
09	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Notion de pas

Nous utilisons aussi la notion de pas qui représente une direction dans le plateau du jeu, le **pas** est en fait combien on ajoute au rang actuel pour qu'on se déplace d'une case dans la direction voulu, le tableau ci-contre résume l'ensemble des pas.

N°	Pas	Sens	Direction
1	-8	↑	Haut
2	-7	↗	Haut, droite
3	1	→	droite
4	9	↘	Bas, droite
5	8	↓	Bas
6	7	↙	Bas, gauche
7	-1	←	Gauche
8	-9	↖	Haut, gauche

Pondération pour la 2^{ème} intelligence artificielle :

Une pondération des cases du plateau est faite pour guider la deuxième intelligence artificielle dans son choix de mouvement (voir la section IA2)

500	-150	30	10	10	30	-150	500
-150	-250	0	0	0	0	-250	-150
30	0	1	2	2	1	0	30
10	0	2	16	16	2	0	10
10	0	2	16	16	2	0	10
30	0	1	2	2	1	0	30
-150	-250	0	0	0	0	-250	-150
500	-150	30	10	10	30	-150	500

Prédicats de Base :

N°	Prédicat	Détail
1	Signature	member (? element, + liste)
	Description	Vérifie si l' element appartient à la liste .
2	Signature	listeMul (+element,+nombre,-liste)
	Description	Renvoie une liste de la forme [element,element,....element] nombre fois. exemple : listeMul(x,3,L) nous donne L=[x , x , x]
3	Signature	longeur (+liste, ?longeur)
	Description	Renvoier dans longeur la taille de la liste
4	Signature	replace (+element, +rang, +liste, +listeResultante)
	Description	la listeResultante reçoit la liste où l'élément du rang « rang » est remplacé par element . exemple : replace(y,2,[x,x,x],L) nous donne L=[x,y,x]
5	Signature	element (?element, ?rang, + Liste)
	Description	vérifie si l'élément du rang rang dans la Liste est bel est bien element . peut être utilisé pour connaître le rang d'un élément ou bien l' élément qui se trouve dans le rang rang . exemple : element(y,2,[x,z,x]) nous donne false .
6	Signature	choose_max_solution (+liste,-solution,-valeur)
	Descripton	A partir de « liste » qui est une liste de couple [X,Y], « valeur » représente le maximum des Y, et « solution » représente le X associé à « valeur »
7	Signature	choose_min_solution (+liste,-solution,-valeur)
	Descripton	A partir de « liste » qui est une liste de couple [X,Y], « valeur » représente le minimum des Y, et « solution » représente le X associé à « valeur »

Prédicats du jeu:

Prédicats globaux

N°	Prédicat	Détail
1	Signature	inverse (+Joueur, -Adversaire)
	Description	Renvoie dans Adversaire l'adversaire du Joueur : Si joueur = blanc alors Adversaire = noir Si joueur = noir alors Adversaire = blanc
2	Signature	getList (+Plateau,+element, -L)
	Description	Renvoie dans la liste L l'ensemble des rangs des éléments element Exemple : getListe([libre, noir,noir],noir,L) nous renvoie L=[2,3]
3	Signature	initP (-Plateau)
	Description	Initialise le Plateau du jeu au début de la partie.
4	Signature	isIndiceLibre (+Indice, +Plateau)
	Description	Vérifie si la cellule n° « Indice » du Plateau est libre.
5	Signature	estNonBorne (+Indice, +Pas)
	Description	Vérifie si on n'a pas dépassé les bornes du plateau en le parcourant dans la direction associée au Pas , et en étant à la cellule n° « Indice ».
6	Signature	isCadrerAvecPas (+PositionCourante, +Joueur, +Pas, +IndiceOrigine, +Indice)
	Description	Ce prédicat vérifie si en posant un pion de Joueur dans la cellule n° « Indice », on va encadrer les pions de l'advesaire sur la direction correspondante au Pas .
7	Signature	direction (+PositionCourante,+Joueur,+Indice,+Pas)
	Description	Ce prédicat lance le premier appel au prédicat isCadrerAvecPas , en lui précisant l'indice de la première cellule sur la direction étudiée.
8	Signature	aEncadrer (+PosCourante,+Joueur, +Indice)
	Description	Renvoie True si le joueur peut poser un pion dans Indice dans la position PosCourante
9	Signature	move (+PosCourante,+Joueur, ?Indice)
	Description	Renvoie dans Indice un des mouvements possibles pour le joueur dans la position PosCourante . si Indice est instancié, elle vérifie si le mouvement est possible.
10	Signature	score (+Plateau, -Blanc, -Noir)
	Description	Retourne le score du joueur blanc et du joueur noir respectivement dans Blanc et Noir .
11	Signature	final (+PlateauCourant, -Gagnant)
	Description	Vérifie si sur le PlateauCourant la partie est finie, et retourne aussi le Gagnant qui peut prendre trois valeurs (noir, blanc, nul).
12	Signature	changer (+PositionCourante, +Joueur, +Indice, -PositionSuivante, +Pas)
	Description	Après qu'un nouveau pion est posé par un Joueur , ce prédicat effectue sur la PositionCourante les changements nécessaires correspondants seulement à la direction associée au Pas , à partir de l' Indice , et retourne le résultat dans PositionSuivante .
13	Signature	maj (+PositionCourante, +Joueur, +Indice, -PositionSuivante, +Pas)
	Description	Après qu'un nouveau pion est posé par un Joueur , ce prédicat vérifie s'il y a des changements à faire sur la direction associée au Pas , et si c'est le cas alors elle les fait à travers le prédicat changer .
	Signature	transform (+PositionCourante, +Joueur, +Indice, -PositionSuivante)
	Description	Après qu'un nouveau pion est posé par un Joueur sur la cellule n° « Indice », le prédicat effectue tous les changements possible dans toutes les directions, et retourne le résultat dans la PositionSuivante .

	Signature	jouer(+PositionCourante, +Joueur, +Indice, -PositionSuivante)
	Description	Vérifie si le Joueur peut poser son pion sur la cellule n° « Indice », si c'est le cas, il effectue tous les changements nécessaires et retourne le résultat dans la PositionSuivante .

Prédicats pour IA1 (Random) :

L'Intelligence Artificiel **IA1** choisit un **mouvement aléatoire** parmi les mouvements possibles

N°	Prédicat	Détail
1	Signature	ia1(+PosCourante,+Joueur,-PosSuivante,-MvtFait)
	Description	Renvoie dans PosSuivant le plateau qui suit PosCourant ,après que l'IA Joueur (noir ou blanc) choisit le mouvement aléatoire retourné dans MvtFait

Affichage sous Prolog

N°	Prédicat	Détail
1	Signature	affichage(+Plateau)
	Description	affiche le Plateau
2	Signature	writeMvtPossible(+Plateau,+Joueur)
	Description	Affiche la liste des mouvements Possible pour le Joueur (noir ou blanc) dans le plateau Plateau

Déroulement Jeu sous Prolog entre IA1 et Humain

N°	Prédicat	Détail
1	Signature	avancerJeu(+Humain,+Plateau,+Mvt)
	Description	prend compte du mouvement choisit Mvt par l' humain (noir ou blanc) et : <ol style="list-style-type: none"> 1. Dans le cas ou le mouvement est non permis, elle rappelle deroulerJeu(Humain,Plateau,Humain) 2. Dans le cas ou le mouvement est permis, elle rappelle deroulerJeu(Humain, PlateauSuivant,IA) en passant la main à l'intelligence artificiel IA1
2	Signature	deroulerJeu(+Humain,+Plateau,+JoueurCourant)
	Description	Ce prédicat affiche le plateau actuel et : <ol style="list-style-type: none"> 1. Si le joueurCourant est l'humain il lit a partir de la console le mouvement est fait appelle à avancerJeu(Humain,Plateau,Mvt) 2. Si le joueurCourant esst l'IA (inverse de Humain) on fait appelle à ia1 <p>Ce prédicat prend fin lorsque le plateau est final est affiche le gagnat, et dans le cas ou pour l'un des deux joeurs aucun mouvement n'est possible, il fait passer la main à l'adversaire</p>
3	Signature	jeuIA1Humain(+Humain)
	Description	lance le jeu Humain contre l'intelligence artificielle ia1 (si Humain est noit le joueur commence sinon l'IA commence)

Prédicats pour IA2

N°	Prédicat	Détail
1	Signature	listePoids(-ListePonderation)
	Description	renvoie la ListePonderation (voir section modélisation)
2	Signature	calculerValeur(+Plateau,+ListePonderation,-Valeur)
	Description	Renvoie dans Valeur la valeur du joueur noir dans le Plateau c'est-à-dire la somme des pondération des cases occupée par le joueur noir
3	Signature	h(+Plateau,-Valeur)
	Description	Renvoie dans Valeur la valeur du joueur noir dans le plateau, si le plateau est en position finale : <ol style="list-style-type: none"> 1. Noir a gagné : Valeur = 5000 2. Blanc a gagné : Valeur = -5000 3. Nul a gagné : Valeur = 0 Si la position n'est pas finale on renvoie dans valeur , CalculerValeur(Plateau,ListePonderation,valeur)
4	Signature	minimax(+Joueur, +PositionCourante, +Valeur, -PositionSuivante, -MouvementFait, +Profondeur)
	Description	Selon l'algorithme de minimax, ce prédicat choisit le mouvement optimal à faire (MouvementFait) pour le Joueur , selon la Profondeur donnée, et retourne la Valeur d'heuristique associée, et le résultat dans la PositionSuivante .
5	Signature	iA2(+PositionCourante, +Joueur, -PositionSuivante, -MouvementFait, +Niveau)
	Description	En utilisant l'heuristique minimax et avec une profondeur « Niveau » qui doit être supérieure ou égale à 1 , ce prédicat choisit le mouvement optimal (MouvementFait) à faire par le Joueur , et retourne le résultat obtenu après ce mouvement dans la PositionSuivante .

Déroulement Jeu sous Prolog entre IA2 et Humain

N°	Prédicat	Détail
1	Signature	avancerJeu2(+JoueurHumain, +Plateau, +Mouvement, +Niveau)
	Description	Vérifie si le Mouvement choisi par le JoueurHumain est valable, si c'est le cas il met à jour le Plateau , et il passe la main à la machine pour qu'elle joue son tour en utilisant iA2 .
2	Signature	deroulerJeu2(+JoueurHumain, +Plateau, +JoueurCourant, +Niveau)
	Description	Déroule le jeu selon le principe suivant : <ul style="list-style-type: none"> • Si le JoueurCourant est humain, alors ce prédicat lui donne la main pour jouer, sinon il effectue un mouvement choisi par l'iA2 selon la profondeur « Niveau »
3	Signature	jeuIA2Humain(+JoueurHumain, +Niveau)
	Description	Lance une partie entre le JoueurHumain et l' iA2 avec une profondeur « Niveau »

Déroulement Jeu sous Prolog entre IA1 et IA2

N°	Prédicat	Détail
1	Signature	deroulerJeuIA1IA2 (+Plateau, +JoueurCourant, +Niveau, -Vainqueur)
	Description	Déroule le jeu entre le noir qui est « IA2 » (avec une profondeur « Niveau ») et le blanc qui est « IA1 », et il nous retourne le Vainqueur à la fin.
2	Signature	jeuIA1IA2 (+Niveau)
	Description	Lance une partie entre IA1 (jouant avec le blanc) et IA2 (jouant avec le noir et avec une profondeur « Niveau »).

Prédicats pour IA3

N°	Prédicat	Détail
3	Signature	h2 (+Plateau,-Valeur)
	Description	Renvoie dans Valeur la valeur du joueur noir dans le plateau, si le plateau est en position finale : <ul style="list-style-type: none"> 4. Noir a gagné : Valeur = 5000 5. Blanc a gagné : Valeur = -5000 6. Nul a gagné : Valeur = 0 Si la position n'est pas finale on renvoie dans valeur , Score du joueur Noir – Score du Joueur Blanc
4	Signature	minimax2 (+Joueur, +PositionCourante, +Valeur, -PositionSuivante, -MouvementFait, +Profondeur)
	Description	Selon l'algorithme de minimax, ce prédicat choisit le mouvement optimal à faire (MouvementFait) pour le Joueur , selon la Profondeur donnée, et retourne la Valeur d'heuristique associée (h2), et le résultat dans la PositionSuivante .
5	Signature	IA3 (+PositionCourante, +Joueur, -PositionSuivante, -MouvementFait, +Niveau)
	Description	En utilisant l'heuristique minimax et avec une profondeur « Niveau » qui doit être supérieur ou égale à 1 , ce prédicat choisit le mouvement optimal (MouvementFait) à faire par le Joueur , et retourne le résultat obtenu après ce mouvement dans la PositionSuivante .

Déroulement Jeu sous Prolog entre IA3 et Humain

N°	Prédicat	Détail
1	Signature	avancerJeu4 (+JoueurHumain, +Plateau, +Mouvement, +Niveau)
	Description	Vérifie si le Mouvement choisi par le JoueurHumain est valable, si c'est le cas il met à jour le Plateau , et il passe la main à la machine pour qu'elle joue son tour en utilisant IA3 .
2	Signature	deroulerJeu4 (+JoueurHumain, +Plateau, +JoueurCourant, +Niveau)
	Description	Déroule le jeu selon le principe suivant : <ul style="list-style-type: none"> • Si le JoueurCourant est humain, alors ce prédicat lui donne la main pour jouer, sinon il effectue un mouvement choisi par l'IA3 selon la profondeur « Niveau »
3	Signature	jeuIA3Humain (+JoueurHumain, +Niveau)
	Description	Lance une partie entre le JoueurHumain et l' IA3 avec une profondeur « Niveau »

Déroulement Jeu sous Prolog entre IA1 et IA3

N°	Prédicat	Détail
1	Signature	deroulerJeuIA1IA3 (+Plateau, +JoueurCourant, +Niveau, -Vainqueur)
	Description	Déroule le jeu entre le noir qui est « IA3 » (avec une profondeur « Niveau ») et le blanc qui est « IA1 », et il nous retourne le Vainqueur à la fin.

Déroulement Jeu sous Prolog entre IA2 et IA3

N°	Prédicat	Détail
1	Signature	deroulerJeuIA2IA3 (+Plateau, +JoueurCourant, +NiveauIA2, +NiveauIA3, -Vainqueur)
	Description	Déroule le jeu entre le noir qui est « IA3 » (avec une profondeur « NiveauIA3 ») et le blanc qui est « IA2 », et il nous retourne le vainqueur à la fin.

Déroulement Jeu sous Prolog entre IA2 et IA2

- Pour voir l'impact de la profondeur sur la même IA.

N°	Prédicat	Détail
1	Signature	deroulerJeuIA2IA2 (+Plateau, +JoueurCourant, +NiveauIA2, +Niveau2IA2, -Vainqueur)
	Description	Déroule le jeu entre le noir qui est « IA2 » (avec une profondeur « NiveauIA2 ») et le blanc qui est « IA2 » (avec une profondeur « Niveau2IA2 »), et il nous retourne le vainqueur à la fin.

Etude de performances des IA :

N°	Prédicat	Détail
1	Signature	testerGagnantEtLancerProchaine (+J1,+J2, +NbrRestant, -Score1, -Score2, +Gagnant)
	Description	Incrémente le score du gagnant (si la partie n'est pas nulle) et fait un appel au predicat qui lancera les prochaines parties.
2	Signature	lancerPlusieursParties (+J1, +J2,+NbrPartie,-ScoreIA1,-ScoreIA2)
	Description	Lance un nombre " NbrPartie " de parties entre J1 et J2 qui peuvent être IA1, IA2, ou IA3, et calcule le score de chacun d'eux.