

RAPPORT JEU OTHELLO

-L'ANNÉE 2014/2015-

NGUYEN TU Nam
SOUPRAMANIANE Sarathbabou

TABLE DE MATIÈRES

I. INTRODUCTION AU JEU :	3
A - INTRODUCTION :	3
B - LES REGLES DU JEU :	3
II. ALGORITHMIQUE DU JEU :	3
1. LES STRUCTURES :	3
A - UN COUP :	3
B - UN OTHELLIER :	3
C - UN JOUEUR :	4
D - UNE PARTIE :	4
2. ALGORITHMES PRINCIPAUX :	4
A - LES FONCTIONS FONDAMENTALES:	4
B- LA FONCTION MAIN() :	4
3. ETAT DES FONCTIONNALITES :	5
4. MODE D'EMPLOI :	5

I. Introduction au jeu :

a - Introduction :

Othello est un jeu de société combinatoire abstrait, qui oppose deux joueurs : Noir et Blanc. Il se joue sur un tablier unicolore de 64 cases, 8 sur 8, appelé othellier. Les colonnes sont numérotées de gauche à droite par les lettres a à h ; les lignes sont numérotées de haut en bas par les chiffres 1 à 8.

Les joueurs disposent de 64 pions bicolores, noirs d'un côté et blancs de l'autre. En début de partie, quatre pions sont déjà placés au centre de l'othellier : deux noirs, en e4 et d5, et deux blancs, en d4 et e5.

Chaque joueur, noir et blanc, pose l'un après l'autre un pion de sa couleur sur l'othellier selon les règles définies ci-dessous. Le jeu s'arrête quand les deux joueurs ne peuvent plus poser de pion. On compte alors le nombre de pions. Le joueur ayant le plus grand nombre de pions de sa couleur sur l'othellier a gagné.

b - Les règles du jeu :

Noir commence toujours la partie[1]. Puis les joueurs jouent à tour de rôle, chacun étant tenu de capturer des pions adverses lors de son mouvement. Si un joueur ne peut pas capturer de pion(s) adverse(s), il est forcé de passer son tour. Si aucun des deux joueurs ne peut jouer, ou si l'othellier ne comporte plus de case vide, la partie s'arrête. Le gagnant en fin de partie est celui qui possède le plus de pions.

La capture de pions survient lorsqu'un joueur place un de ses pions à l'extrémité d'un alignement de pions adverses contigus et dont l'autre extrémité est déjà occupé par un de ses propres pions. Les alignements considérés peuvent être une colonne, une ligne, ou une diagonale. Si le pion nouvellement placé vient fermer plusieurs alignements, il capture tous les pions adverses des lignes ainsi fermées. La capture se traduit par le retournement des pions capturés. Ces retournements n'entraînent pas d'effet de capture en cascade : seul le pion nouvellement posé est pris en compte.

II. Algorithmique du jeu :

1. Les structures :

BLANC = 1 et NOIR = 0

Ce choix de valeur pour les couleurs est extrêmement pratique car il correspond aux valeurs des indices des joueurs. Ainsi, dans le programme, l'indice d'un joueur est une couleur et une couleur est l'indice d'un joueur ! Par ailleurs dans l'algèbre de Boole 0 = NON 1.

Remarque, du fait de cet tri-équivalence, et par abus de langage, on utilisera le terme allier et adversaire pour désigner un joueur ou une couleur d'un pion en référence à une couleur représentative ou, un joueur représentatif (autrement dit dans un certain contexte d'une fonction).

VIDE = 2 et BORD = 3

Ces deux nouvelles constantes concernent l'othellier à part entière. Une case peut être vide (VIDE) et si on va trop loin sur l'othellier on arrive sur un bord (BORD). La pertinence de cette remarque va naître dans l'explication de la structure de l'othellier. De même, le choix de la valeur de VIDE sera expliqué dans la partie des patterns.

a - Un coup :

Un coup peut être représenté par le joueur qui joue ce coup, autrement dit la couleur du pion joué puisque l'implémentation des couleurs et des joueurs est la même dans ce programme... Il est important d'y ajouter la position de la case jouée. De plus, d'un point de vue contextuel, un vecteur donnant de nombre de retournement pour chaque direction.

Les deux premiers champs étant totalement évident, notons plutôt l'idée de retournement. Cette façon d'implémenter la structure va permettre l'annulation d'un coup. En effet, pour un othellier, une position et une couleur donnés : il est impossible de reconstruire l'othellier tel qu'il été avant le dernier coup ! L'information du nombre de pions à retourner pour toutes les directions permet cette manipulation.

b - Un othellier :

Part définition un othellier est un plateau de jeu de 64 cases (8x8). Pour s'affranchir de nombreuses contraintes d'existence d'indices, on choisi un vecteur de 100 cases. Dans ce cas de figure 36 cases seront donc présentatrice des bords de l'othellier. Cette représentation permet de considérer que deux nouvelles couleurs existes : le vide et le bord. Quand, par exemple, on se déplace dans une certaine direction depuis une position, on devra s'arrêter si l'on est sur un bord : Un seul test sera fait et il ne dépendra que du contenu de la case. Cette

représentation est donc très avantageuse, toutefois on ajoutera à la structure un petit vecteur de deux cases contenant chacune le nombre de pion de chaque joueur. Cet ajout permettra d'éviter de recompter tous les pions systématiquement.

c - Un joueur :

Le joueur est une structure très simple. Comme dans la vie de tous les jours on attribue un nom à chaque personne. C'est ce qui va être fait pour cette structure. En outre, en petit plus doit intervenir du fait de l'intelligence artificielle : Le type. Le type d'un joueur est une énumération des niveaux de difficultés. Il est nul si le joueur est humain, autrement, il codera la difficulté de l'intelligence artificielle (de 1 à 3 car il a été choisi de faire trois niveaux de difficulté). Remarque, l'état du code source actuel permet de réaliser neuf niveaux.

d - Une partie :

La partie sera la structure la plus complexe du programme. Elle se composera de deux joueurs, d'un othellier, et de 60 coups. Il serait judicieux d'y ajouter une valeur qui porterait la couleur du joueur qui doit jouer. Cette structure à l'air complète, toutefois, au cours de la programmation, il sera question de ce qui se situe dans le vecteur des coups, en d'autres termes, de connaître à quel coup on se situe. Pour ce faire, il va falloir introduire les valeurs traditionnelles d'une boucle, soit, un incrément que l'on nommera état et une valeur maximale que l'on nommera longueur.

2. Algorithmes principaux :

a - Les fonctions fondamentales:

** legal() :*

Entrée :

- Othellier oth, coup c, variable num
- Variables i, j, legal

Sortie :

Si le coup est légal, retourne à 1, sinon 0

Début :

- Si la position de la case est VIDE ou n'est pas dans l'othellier ; retourne à 0 ;
- Si c'est le tour du 1^{er} joueur (NOIR) ;
Vérifier 8 directions, si le coup est légal, la variable « legal » = 1 ;
- Si c'est le tour du 2^{me} joueur (BLANC) ;
On fait la même chose ;

Fin :

Retourne à « legal »

** jouer()*

Entrée :

- Othellier oth, coup c, int num
- Variable i, k, temp, pos

Sortie :

Othellier oth

Début :

- Si c'est le tour du 1^{er} joueur (NOIR) ;
Si le pion est bien placé dans l'othellier, on teste les 8 directions comme dans la fonction legal() et change la couleur des pions BLANC en NOIR ;
- Si c'est le tour du 2^{me} joueur (BLANC) ;
On fait la même chose ;

Fin :

Othellier oth ;

b- La fonction main() :

Entrée :

- partie p
- variable num

Sortie :

Retourne à 0

Début :

```

Initialisation l'othellier ;
Affiche l'othellier ;
Initialisation le numéro de joueur ;
Tant que la partie n'est pas encore finie ;
    Le joueur choisi le coup ;
    Si le coup est légal ;
        Le joueur peut jouer ce coup ;
        Affiche l'othellier ;
    Après on change le tour de joueur ;
Fin :
    Retourne à 0 ;

```

3. *Etat des fonctionnalités :*

Fonction	A	B	C	D	E	Commentaire
init()		X	X	X	X	Initialisation l'othellier
affiche_othellier()		X	X	X	X	Affichage l'othellier
bloque()		X	X	X	X	Vérification qui est le gagnant du jeu
legal()		X	X	X	X	Vérification le coup s'il est légal ou pas
change()		X	X	X	X	Changement du joueur
choose()		X	X	X	X	Permet le joueur de choisir un coup
peut()		X	X	X	X	Vérification si le joueur peut jouer encore
jouer()		X	X	X		Permet le joueur de jouer un coup
nom()		X				Permet le joueur d'insérer son nom et prénom

En construire au fur à mesure...

A : Non réalisé
B : Spécifié
C : Codé
D : Testé
E : Consolidé

4. *Mode d'emploi :*

En construire au fur à mesure...