Quelques structures de données pour Othello

1 Objectif de ce document

Ce document propose des structures de données assez pratiques d'utilisation pour la programmation d'un jeu d'Othello. Les deux structures les plus importantes sont la structure `othellier` et (surtout) la structure `coup`. Comme présenté dans un document précédent, il est assez pratique, même si éventuellement contre-intuitif au premier abord, de représenter un othellier par un tableau à une seule dimension avec des cases supplémentaires modélisant le bord (extérieur) du damier : la navigation sur l'othellier est facilitée par l'utilisation d'un tableau de directions, donc chacune contient la valeur de l'incrément à ajouter à la position courante pour se déplacer dans la direction souhaitée (ces directions correspondent aux constantes NO, N, NE, etc, du fichier `constantes.h`). Un coup est caractérisé par une position, une couleur et un tableau de retournements dans les huit directions possibles. La connaissance de ce tableau, qui indique donc le nombre de pions à retourner dans chacune des huit directions lorsque l'on joue à un certain endroit de l'othellier contient *toute* l'information nécessaire à l'annulation d'un coup.

2 Constantes et structures de données

Le Listing 1 donne un exemple de constantes qui peuvent être utiles pour le programme. Le Listing 2 propose des structures de données suffisantes pour la programmation du jeu. Vous n'êtes bien évidemment pas obligés d'utiliser ces structures de données mais elle peuvent éventuellement vous donner des idées si vous êtes en manque d'inspiration.

Listing 1 – constantes.h

```
/**************************************************************************
                        constantes.h  -  description
                             -------------------
    creation             : wed feb 20, 2008
    copyright            : (C) 2008 by Liva Ralaivola
    email                : liva.ralaivola@lif.univ-mrs.fr
**************************************************************************/

/**************************************************************************
 *                                                                        *
 *   This program is free software; you can redistribute it and/or modify *
 *   it under the terms of the GNU General Public License as published by *
 *   the Free Software Foundation; either version 2 of the License, or    *
 *   (at your option) any later version.                                  *
 *                                                                        *
 **************************************************************************/


/**************************************************************************
 *   Définition de quelques constantes de base servant à plusieurs        *
 *   endroits dans le programme.                                          *
 **************************************************************************/

#ifndef __CONSTANTES__
#define __CONSTANTES__

/** Tableau uni-dimensionnel de 100 cases
    pour la représentation d'un damier 8x8 */
#define SIDE        8
```

```
#define LONG_SIDE   10
#define MAX_INDEX   100
#define DIRECTIONS  8

/** Les 8 directions a scruter */
#define NO  -11
#define N   -10
#define NE   -9
#define O    -1
#define E     1
#define SO    9
#define S    10
#define SE   11

/** Ces choix de NOIR et BLANC permettent
    de passer d'une couleur à la couleur
    adverse très facilement                */
#define JOUEURS 2
#define NOIR     0
#define BLANC   +1
#define VIDE    +2
#define BORD    -1

/** Autres constantes dépendant de l'implémentation */
#define MAX_MOBILITE 30
#define MAX_COUPS    60
#define MAX_STRING   256
#define MAX_MATERIEL 64

#define HUMAIN     0
#define MACHINE    1

#define UNSET   -99
#endif
```

Listing 2 – structures.h

```
/*************************************************************************
                        structures.h  -  description
                             ------------------
    creation             : wed feb 20, 2008
    copyright            : (C) 2008 by Liva Ralaivola
    email                : liva.ralaivola@lif.univ-mrs.fr
*************************************************************************/

/*************************************************************************
 *                                                                       *
 *   This program is free software; you can redistribute it and/or modify  *
 *   it under the terms of the GNU General Public License as published by  *
 *   the Free Software Foundation; either version 2 of the License, or     *
 *   (at your option) any later version.                                   *
 *                                                                       *
 *************************************************************************/


/*************************************************************************
```

```
 *   Définition des structures de base pour la programmation d'un jeu      *
 *   d'Othello. Les types pointeurs sont également déclarés afin de        *
 *   l'allègement du code lors du passage d'arguments par adresse          *
 ************************************************************************/
#ifndef __STRUCTURES__
#define __STRUCTURES__

#include "constantes.h"

/**
 *  Un coup est décrit par une position, une couleur et le nombre
 *  de pions retournés dans chacune des 8 directions. On peut alors
 *  facilement annuler un coup.
 */
typedef struct coup *coup;
struct coup{
  int position;
  int couleur;
  int retournement[DIRECTIONS];
};

/**
 *  Un Othellier est décrit par un damier et le nombre
 *  de pions noirs et de pions blancs sur l'othellier.
 *  A chaque opération sur l'othellier, ces tableaux doivent
 *  être maintenus à jour.
 */
typedef struct othellier *othellier;
struct othellier{
  int damier[MAX_INDEX];
  int materiel[JOUEURS];
};

/**
 *  Un joueur: un nom, un type et un niveau (pour les ordinateurs)
 */
typedef struct joueur *joueur;
struct joueur{
  char nom[10];
  int  type;    // 0 si humain et 1 si ordinateur
  int  niveau;
};

/**
 *  Une partie: un ensemble de positions de jeu pour deux joueurs
 *  Le curseur spécifie l'endroit de la partie où l'on se situe (si,
 *  par exemple, on est dans un phase d'analyse d'une partie).
 *  Le trait est la couleur du joueur devant jouer le prochain coup.
 */
typedef struct partie *partie;
struct partie {
  int         curseur;
  int         trait;
  int         longueur;
  int         final;
  joueur      joueur[2];
```

```
  struct coup coup[MAX_COUPS];
  othellier   othellier;
};



#endif
```

3 Un fichier header pour bien démarrer

Le Listing 3 donne un exemple de fichier header qui déclare les fonctions de gestion d'un othellier. La définition de ces différentes fonctions doit permettre très rapidement d'avoir un programme exécutable fonctionnel pour que deux humains jouent l'un contre l'autre.

Listing 3 – othellier.h

```
/**************************************************************************
                          othellier.h  -  description
                             ------------------
    creation             : wed feb 20, 2008
    copyright            : (C) 2008 by Liva Ralaivola
    email                : liva.ralaivola@lif.univ-mrs.fr
**************************************************************************/

/**************************************************************************
 *                                                                        *
 *   This program is free software; you can redistribute it and/or modify *
 *   it under the terms of the GNU General Public License as published by *
 *   the Free Software Foundation; either version 2 of the License, or    *
 *   (at your option) any later version.                                  *
 *                                                                        *
 **************************************************************************/

/**************************************************************************
 *                                                                        *
 *   Les prototypes de fonctions pour la gestions d'othellier             *
 *                                                                        *
 **************************************************************************/

#ifndef __OTHELLIER__
#define __OTHELLIER__

#include "structures.h"

/** Initialise l'othellier en position de départ */
void initialiser (othellier);

/*************************************************************
 * Renvoie 1 si le coup décrit par une couleur et une position     *
 * est légal et remplit le tableau de retournement s'il le faut    *
 * et renvoie -1 sinon. Le dernier paramètre permet de spécifier   *
 * si l'on souhaite une recherche complète de coup légal ou pas :  *
 * n'est utile que pour la programmation de l'ia.                  *
*************************************************************/
int  legal       (othellier, coup, int);
```

```
/** Joue le coup, qui est légal */
void joue_coup   (othellier, coup);

/** annule le coup, qui est légal */
void annule_coup (othellier, coup);

/** retourne vrai si le joueur de la couleur en question peut jouer */
int peut_jouer(othellier, int);

/** retourne vrai si l'othellier est bloqué, i.e. fin de partie */
int bloque(othellier);

/** retourne la couleur du joueur ayant le trait etant
    donné le dernier joueur à avoir joué */
int trait(othellier, int);

#endif
```