Initiation à la Programmation C
Devoir maison - L2
Attaxx (simplifié)

Nous nous proposons d'implémenter en `C` une variante du *reversi*, aussi appelé *othello*.

1 Le jeu Attaxx

Le plateau de jeu est une grille carrée de taille `N x N`, où `N` est un entier définit en tête de fichier qui vaut usuellement `7`. Il contient au début deux pions noirs et deux pions blancs placés sur des coins oppposés du plateau, comme sur le plateau de la figure suivante :

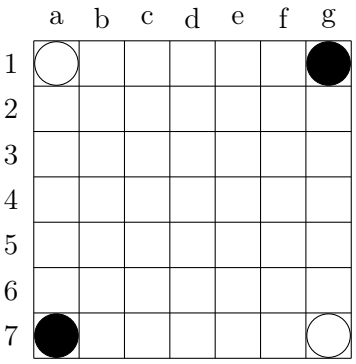
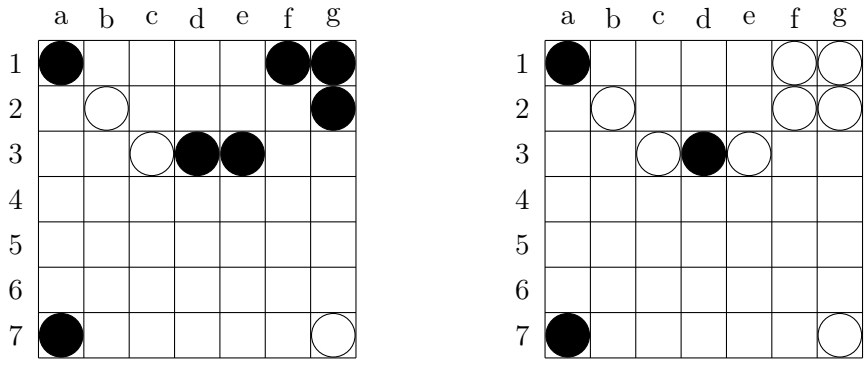


Figure 1: Un exemple de plateau initial

Un coup consiste à placer un pion de sa couleur sur une des huit cases voisines d'un pion adverse déjà posé. La case doit être vide pour que l'on puisse poser un pion dessus. De plus, après avoir posé un pion, tous les pions adverses sur les cases voisines de la case du pion posé changent de couleur en devenant de la couleur du pion posé.

Un exemple de coup valide est donné par la figure suivante. Le joueur joue en $(2, f)$. Notons bien qu'il n'aurait pas pu jouer en $(2, c)$, n'étant pas une case voisine d'un pion adverse.



Avant le coup — Après avoir joué en $(2, f)$.

Figure 2: Un exemple de plateau initial

Le jeu se termine lorsque le plateau de jeu est rempli ou lorsqu'un joueur ne possède plus de pions. Le gagnant est alors celui qui a le plus de pions sur le plateau.

2 Implémentation

Pour implémenter ce jeu, nous nous donnons tout d'abord deux constantes :

```
#define TAILLE_PLATEAU 7
#define TAILLE_MAX_NOM 20
```

Nous utiliserons deux structures, une pour modéliser des `Joueurs`, une autre pour modéliser le `Plateau` de jeu.

2.1 Les `Joueurs`

La structure `Joueur` sera :

```
typedef struct {
    char nom[TAILLE_MAX_NOM];
    char symbol;
    int score;
} Joueur;
```

A chaque `Joueur` est associé un symbol (typiquement, `'x'` et `'o'`) pour faciliter le développement en mode textuel. Son score correspond au nombre de pion qu'il possède actuellement sur le plateau.

2.2 Le `Plateau`

La structure `Plateau` sera définie par :

```
typedef struct {
    char plateau[TAILLE_PLATEAU + 2][TAILLE_PLATEAU + 2];
    Joueur * joueurs[2];
} Plateau;
```

Les indices valides du plateau sont `1`, `2`, ..., `TAILLE_PLATEAU`. Autour du plateau, on trouvera alors le cractère `'*'` pour symboliser des cases virtuelles. Une case vide du plateau de jeu contiendra, elle, le caractère `'.'`.

N.B. 1 : Le fait d'entourer le plateau de jeu de cases virtuelles vides permettra de parcourir plus simplement les huit cases voisines d'une case se situant sur le bord du plateau.

N.B. 2 : La structure `Plateau` contient un tableau de pointeurs sur les structures `Joueur` pour éviter de recopier intégralement les structures `Joueur` sur la pile d'appel. Seul les adresses seront alors recopiées.

3 Consignes

Votre rendu permettra de produire un exécutable dont le comportement dépendra des paramètres transmis sur la ligne de commande. Il acceptera les options:

- `-a` pour indiquer que l'affichage sera en ASCII (option excluant l'option suivante) ;
- `-g` pour indiquer que l'affichage sera graphique (option exclu l'option précédente) ;
- `-h` pour indiquer que deux joueurs humains joueront (option excluant l'option suivante) ;
- `-o` pour indiquer qu'un joueur humain jouera contre l'ordinateur (option exclu l'option précédente).

Chaque fonction sera documentée, avec à minima une description de

1. la fonction ;
2. ses arguments ;
3. sa valeur de retour ;
4. ses éventuels effets de bords avec éventuellement un exemple

L'ordinateur, lorsqu'il jouera, jouera de manière aléatoire à un endroit licite.

3.1 Version ASCII (12 points)

Pour réaliser la version ASCII, on demande :

- d'écrire des fonctions d'initialisation des structures nécessaires à l'implémentation ;
- de réaliser la pose d'un pion sur le plateau de jeu, en respectant les règles énoncées précédemment ;
- de gérer les conditions de fin de partie ;
- de respecter scrupuleusement le formatage du fichier `exemple_sortie.txt` pour réaliser l'affichage ASCII. En particulier, le score courant sera affiché après chaque coup joué. En fin de partie, le gagnant sera annoncé de la manière suivante :

  ```
  Bravo Shaibel, vous avez gagné 30 à 19.
  ```

  ou alors en cas d'égalité par :

  ```
  Egalité 18 à 18.
  ```

- de gérer correctement les options `a`, `g`, `h` et `o`.

3.2 Version graphique (5 points)

En utilisant la librairie graphique `MLV`, le plateau de jeu sera dessiné. Le nom du joueur courant sera affiché. Les clics de la souris remplaceront la saisie des coordonnées où le joueur courant souhaite jouer.

3.3 Améliorations (3 points)

Plusieurs améliorations peuvent être proposées :

- Permettre de poser un pion sur le plateau s'il est à distance 1 ou 2 d'un pion adverse, avec changement de couleur lorsque l'on est à distance 1 uniquement *(facile)* ;
- Jouer avec un plateau contenant des zones particulière (des trous où l'on ne peut pas jouer, des refuges anti contamination, etc.) *(facile)* ;
- Faire jouer l'ordinateur de "manière intelligente" *(de difficile à très difficile)* ;
- Contaminer un voisin avec une probabilité fixé à l'avance (0.75 par exemple) *(facile)* ;
- etc.

Faites vous plaisir ! Mais...

Attention : Les améliorations ne seront prises en compte que si la version ASCII et la version graphique fonctionnent parfaitement.

```
Quel est le nom du premier joueur (symbol o) : Beth
Quel est le nom du second joueur (symbol x) : Shaibel

* * * * *
* x . o *
* . . . *
* o . x *
* * * * *

Beth (o), veuillez saisir les coordonnées où jouer (entre 1 et 3) : 0 1
Beth (o), veuillez saisir les coordonnées où jouer (entre 1 et 3) : 1 2
* * * * *
* o o o *
* . . . *
* o . x *
* * * * *
Score actuel : Beth$(o) 4 - Shaibel(x) 1

Shaibel (x), veuillez saisir les coordonnées où jouer (entre 1 et 3) : 1 1
Shaibel (x), veuillez saisir les coordonnées où jouer (entre 1 et 3) : 2 2
* * * * *
* x x x *
* . x . *
* x . x *
* * * * *
Score actuel : Beth(o) 0 - Shaibel(x) 6

Bravo Shaibel, vous avez gagné 6 à 0.
```

Figure 3: Contenu du fichier `exemple_sortie.txt`.

4 Annexe

Attention : L'affichage ASCII d'un tableau contient un espace invisible en fin de chaque ligne. Ne l'oubliez pas, cela est *extrêmement* important pour la correction automatique !