

Listes circulaires : Jeu de Mancala

Le but de cette séance de Travaux Pratiques est d'implémenter le jeu de Mancala en utilisant des listes circulaires.

Vous trouverez sur Moodle une archive contenant un certain nombre de fichiers pour vous aider à démarrer. Après l'avoir téléchargée :

```
$ tar -xf sda1-tp4.tar.gz  
$ cd sda1-tp4
```

Le fichier `Makefile` contient les règles nécessaires pour compiler deux programme intitulés `test_liste` et `jeu`.

```
$ make  
$ ./test_liste
```

Le programme `test_liste` vous aidera à tester votre implémentation des listes circulaires. Vous pouvez le modifier pour ajouter les tests que vous jugez pertinents.

```
$ make  
$ ./jeu
```

Le programme `jeu` correspond au jeu de Mancala. Lors de l'exécution de ce programme, les flèches `<gauche>` et `<droite>` permettent de sélectionner un champ à égrainer tandis que la touche `<espace>` indique qu'il faut égrainer le champ sélectionné. La touche `` indique qu'il faut brûler le champ sélectionné (ainsi que le champ opposé).

Règles du jeu de Mancala

Règles basiques

Le plateau de jeu est composé de 2 rangées de 6 *champs* (ou *trous*) qui contiennent des *graines* auxquels s'ajoutent deux *champs* spéciaux : les *greniers* (ou *maisons*) des joueurs. Chaque joueur possède l'une des deux rangées. Le score d'un joueur correspond au nombre total de *graines* se trouvant dans les *champs* de sa rangée.

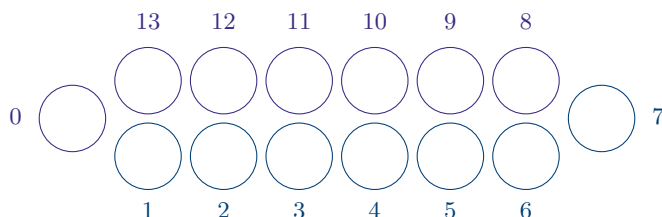
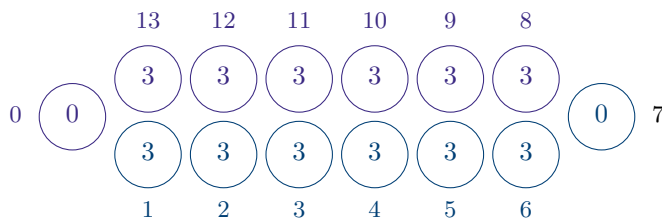


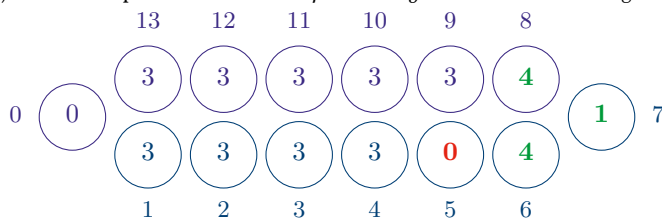
FIGURE 1 – Numérotation des *champs* du plateau de Mancala

Si l'on numérote les *champs* en partant de la gauche dans le sens inverse des aiguilles d'une montre (voir également Figure 1) :

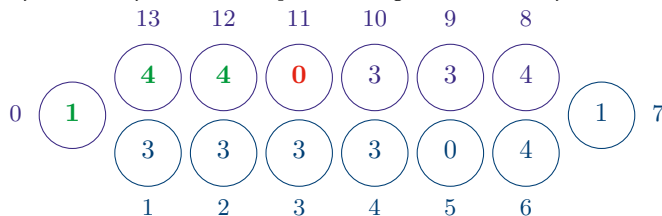
- le *champ* 0 est le *grenier* du **joueur 2** et le *champ* 7 est le *grenier* du **joueur 1**.
- les *champs* 1 à 6 sont les *champs* du **joueur 1**.
- les *champs* 8 à 13 sont les *champs* du **joueur 2**.



(a) Début de la partie. Tous les *champs* sauf les *greniers* contiennent 3 graines.



(b) Le **joueur 1** a égrainé le *champ* 5 : le *champ* 5 ne contient plus aucune graine et les *champs* 6, 7 et 8 ont une graine supplémentaire.



(c) Le **joueur 2** a égrainé le *champ* 11 : le *champ* 11 ne contient plus aucune graine et les *champs* 12, 13 et 0 ont une graine supplémentaire.

FIGURE 2 – Exemple de déroulement

En début de partie, tous les *champs* excepté les *greniers* contiennent 3 *graines*. Chacun son tour, les joueurs *égrainent* (ou *déplacent*) des *graines* depuis un *champ* de leur choix selon les règles suivantes :

- le déplacement se fait dans le sens inverse des aiguilles d'une montre.

- le déplacement vide le *champ* de départ.
- les *graines* sont réparties à raison d'une *graine* par *champ* ou *grenier* situé après le *champ* de départ.
- un joueur ne peut pas initier un déplacement à partir d'un *grenier*.
- un joueur ne peut pas initier un déplacement à partir d'un *champ* de la rangée de son adversaire.

Par conséquent, les *graines* se trouvant dans un *grenier* appartiennent définitivement au joueur correspondant tandis que les autres *graines* sont susceptibles de passer chez l'adversaire si un déplacement implique une quantité trop élevée de *graines*.

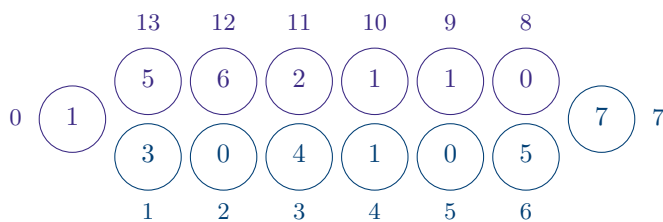
La partie s'arrête lorsqu'au moins l'un des deux joueurs ne peut plus jouer.

Règles avancées

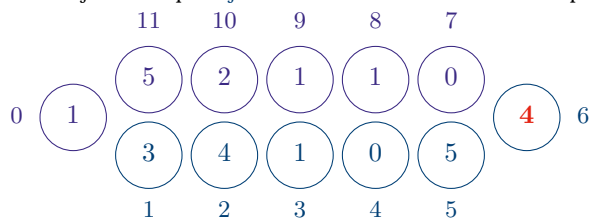
Le jeu décrit précédemment est une version très basique du jeu de Mancala. La version la plus connue comporte les règles supplémentaires suivantes :

- si la dernière *graine* déposée est déposée dans un *grenier*, le joueur rejoue.
- si la dernière *graine* déposée est déposée dans un *champ* vide et si le *champ* opposé contient des *graines*, toutes ces *graines* sont déplacées dans le *grenier* du joueur.

Enfin, voici un coup alternatif qui peut pimenter les parties : l'incendie de champs. Un joueur peut décider de sacrifier 3 graines de son *grenier* (à condition d'avoir au moins 3 dans son *grenier*) pour brûler une colonne de champs à condition que le nombre de champs par joueur soit supérieur à 3. Ces champs deviennent inutilisables et sont enlevés du plateau de jeu.



(a) Plateau de jeu avant que le joueur 1 ne brûle la colonne de champs 2–12.



(b) Le joueur 1 a brûlé les champs 2 et 12 : le plateau ne contient plus que 12 champs et le grenier du joueur 1 a perdu 3 graines.

FIGURE 3 – Exemple d'incendie de champs

1 Implémentation des listes circulaires

Vous devez dans un premier temps implémenter les listes circulaires. La structure retenue est similaire à la structure utilisée pour des listes chaînées classiques :

```
struct liste
{
    /** \brief Feves. */
    unsigned int feves;
    /** \brief Élément suivant dans la liste. */
    struct liste* suivant;
};
```

La différence avec les listes classiques réside dans le fait que dans une liste circulaire, tous les maillons, y compris le dernier maillon, pointent vers un autre maillon de la chaîne (voir Figure 4).

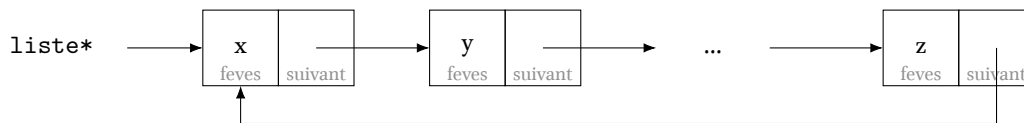


FIGURE 4 – Représentation graphique d'une liste circulaire

2 Logique de base du Jeu de Mancala

Une fois les listes implémentées, il faut implémenter la logique du jeu. La structure proposée est la suivante :

```
struct mancala
{
    liste* plateau;
    unsigned int joueur;
    size_t selection;
};
```

Le champ `joueur` sert à déterminer quel joueur doit jouer. Le comportement de certaines fonctions doit changer selon la valeur de ce champ :

```
bool mancala_coup_valide(const mancala* m, size_t i);
unsigned int mancala_joueur(const mancala* m);
mancala* mancala_egrainer(mancala* m, size_t i);
```

Par exemple, si c'est au joueur 1 de jouer, `mancala_coup_valide()` doit toujours retourner `false` pour le *champ* 13 (c'est un *champ* du joueur 2) et retourner `true` pour le *champ* 2 si, et seulement si, celui-ci contient des graines (sinon il faut retourner `false`). Autre exemple, `mancala_egrainer()` doit changer la valeur du champ `joueur` et réinitialiser le champ `selection` à une valeur adéquate pour le joueur.

Le champ `selection` est utilisé pour stocker la sélection du joueur dans l'interface :

```
size_t mancala_selection(const mancala* m);
mancala* mancala_selection_gauche(mancala* m);
mancala* mancala_selection_droite(mancala* m);
```

Il faut veiller à ce qu'un joueur ne puisse pas faire une sélection dans la rangée de son adversaire.

Note : Les règles stipulent que les *greniers* ne contiennent pas de *graines* au démarrage de la partie. Pour commencer, vous pouvez initialiser tous les *champs* (y compris les *greniers*) de la même manière (ça ne change pas le score des joueurs).

3 Fonctionnalités supplémentaires

1. Modifier le jeu pour supporter les règles avancées.
2. Implémenter l'incendie de *champs* dans les fonctions suivantes :

```
bool mancala_peut_bruler(const mancala* m);  
mancala* mancala_bruler(mancala* m, size_t i);
```

A Fuites mémoires

Pour rappel ou information, vous pouvez vérifier l'absence de fuites mémoires à l'aide d'outils tels que `valgrind(1)` (voir Section A.1) ou bien en activant des options d'instrumentation (voir Section A.2).

A.1 Valgrind

Compilez vos programmes avec les informations de *debugging*, puis exécutez `valgrind(1)` en passant en paramètre le programme cible et ses options :

```
$ make -B CFLAGS="-Og -ggdb"  
$ valgrind ./jeu  
$ valgrind --leak-check=full ./jeu
```

A.2 Instrumentation

Compilez vos programmes en ajoutant l'option d'instrumentation `-fsanitize=address` puis exécutez votre programme de la manière classique :

```
$ make -B CFLAGS="-fsanitize=address" LDFLAGS="-fsanitize=address"  
$ ./jeu
```