Un jeu de Nim

1 Description du jeu

Deux joueurs s'affrontent autour d'un damier de N lignes et M colonnes. Au début du jeu, R pions ($1 \leq R \leq N$) sont positionnés dans la première colonne du damier, de la case de coordonnées $(1, 1)$ jusqu'à la case de coordonnées $(1, R)$. La figure 1 ci-dessous montre l'état initial d'un damier pour $N = 5$, $M = 5$ et $R = 4$.

À tour de rôle, chaque joueur choisit librement un pion et le déplace, dans les limites du damier, soit d'une ou deux cases vers la droite, soit d'une ou deux cases vers le bas. Pendant la partie, le chevauchement d'un autre pion est autorisé et il est possible que plusieurs pions occupent une même case.

Dans la suite, on appellera **voisine** toute case accessible depuis une case donnée. Compte tenu des déplacements possibles, chaque case possède donc 1, 2, 3 ou 4 voisines selon sa position sur le damier (voir figure 1). Une exception cependant : la case (N, M) qui ne possède aucune voisine. On dira que cette case est le **puits** du damier.

La règle du jeu est telle que tout pion termine immanquablement sa course sur la case (N, M) où il se retrouve bloqué. Lorsqu'un pion arrive sur cette case, il est sorti du jeu : on dira que le pion est tombé dans le puits.

Le gagnant est celui qui aura fait tomber le dernier pion dans le puits ; le perdant est donc celui qui ne peut plus jouer parce qu'il n'y a plus de pions sur le damier.

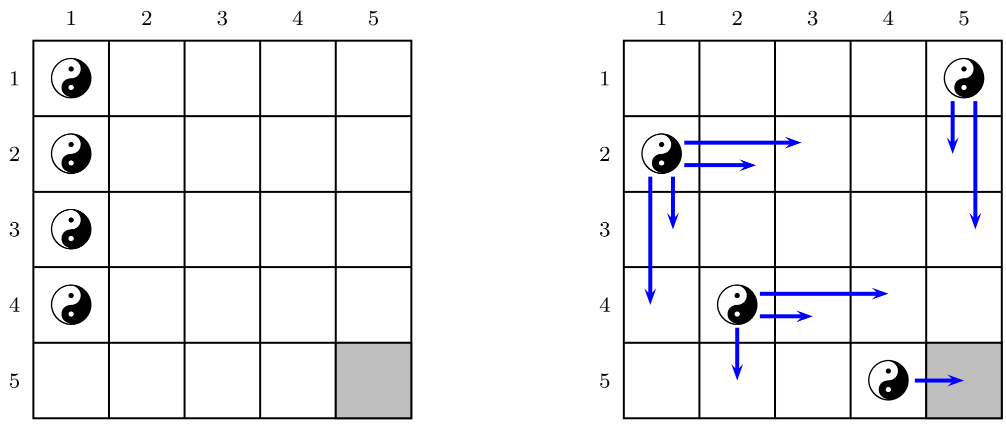

Figure 1 – Etat initial du damier et déplacements possibles en cours de partie

Ce jeu appartient à la famille des jeux de **Nim** dont la principale caractéristique est qu'il est toujours possible de déterminer une stratégie gagnante, soit pour le joueur qui commence la partie, soit pour celui qui joue en second (cela dépend des dimensions du damier et du nombre initial de pions). Cette stratégie est détaillée dans les paragraphes 3 et 4.

2 Travail demandé

Il s'agit d'écrire un programme permettant de gérer une partie opposant un joueur à l'ordinateur. Les paramètres du jeu seront saisis par le joueur : le nombre N de lignes, le nombre M de colonnes, le nombre R de pions initialement présents sur le damier, le niveau de difficulté du jeu et enfin qui, de l'ordinateur ou du joueur, commencera la partie.

La stratégie gagnante pourra être utilisée par l'ordinateur dans les limites fixées par le niveau de difficulté qui variera de (1) à (3) :

(1) **Niveau débutant** : l'ordinateur jouera un coup au hasard avec la probabilité *0.9* ou un coup gagnant avec la probabilité *0.1*.

(2) **Niveau moyen** : l'ordinateur jouera un coup au hasard avec la probabilité *0.5* ou un coup gagnant avec la probabilité *0.5*.

(3) **Niveau expert** : l'ordinateur jouera un coup au hasard avec la probabilité *0.1* ou un coup gagnant avec la probabilité *0.9*.

Jouer au hasard consiste tout d'abord à choisir au hasard un pion sur le damier, puis à choisir au hasard un déplacement autorisé pour ce pion (si le hasard fait bien les choses, un coup au hasard peut être un coup gagnant). Pour jouer un coup gagnant, l'ordinateur utilisera la stratégie gagnante. S'il n'existe aucun coup gagnant, l'ordinateur jouera alors un coup au hasard.

Par ailleurs, le programme devra veiller au respect des règles du jeu et ne devra s'arrêter qu'en fin de partie, après avoir désigné le vainqueur.

Exercice d'analyse
Considérons une partie avec un seul pion sur un damier à 3 lignes et à 3 colonnes. Déterminer une stratégie gagnante et indiquer lequel des deux joueurs pourra l'utiliser.

3 Stratégie gagnante en présence d'un seul pion

3.1 Le nimber

Considérons un damier et un unique pion en case (1,1). Nous allons marquer (ou numéroter) les cases du damier avec des entiers naturels à l'aide de l'algorithme suivant :

(i) Numéroter 0 le puits.

(ii) Sélectionner une case non marquée c dont les voisines sont toutes marquées. Numéroter ensuite c avec le plus petit entier naturel n'apparaissant pas dans les cases voisines.

(iii) S'il reste des cases non marquées, retourner à l'étape *(ii)*.

D'un point de vue pratique, on commence par numéroter, de bas en haut, les cases de la dernière colonne : de cette façon on pourra marquer chaque case conformément à l'étape *(ii)* de l'algorithme. On passe ensuite à l'avant-dernière dernière colonne et on numérote les cases de bas en haut, et ainsi de suite jusqu'à la première colonne du damier (voir figure 2), la dernière case numérotée étant la case (1,1). Puisque toute case possède au plus 4 voisines, les 4 entiers 0, 1, 2 et 3, suffisent pour numéroter les cases d'un damier de taille quelconque. Ces numéros sont appelées **nim-number** ou **nimber**.

	1	2	3	4	5
1	0	1	3	0	1
2	1	0	2	1	0
3	3	2	0	3	2
4	0	1	3	0	1
5	1	0	2	1	0

Figure 2 – Les nimbers d'un damier à 5 lignes et 5 colonnes

Exercice
Calculer les nimbers pour un damier à 7 lignes et 10 colonnes et observer attentivement le résultat obtenu. En déduire une fonction qui calcule le nimber d'une case à partir de ses coordonnées, du nombre de lignes N et du nombre de colonnes M. **Quel est l'intérêt de cette fonction ?**

3.2 La stratégie gagnante

La position nulle
Le but du jeu est d'atteindre le puits dont le nimber est fixé à 0. D'autres cases du damier ont aussi un nimber égal à 0. On dira qu'un pion est en **position nulle** s'il occupe une case de nimber 0 et qu'il est en **position non nulle** s'il occupe une case de nimber différent de 0.

Propriétés

(1) Par construction, toute voisine d'une case de nimber nul possède un nimber non nul : **un pion en position nulle ne peut être déplacé que vers une position non nulle**.

(2) Par construction, toute case de nimber non nul possède au moins une voisine de nimber nul : **il est toujours possible de déplacer un pion d'une position non nulle vers une position nulle**.

La stratégie
Compte tenu de ces propriétés, pour gagner la partie un joueur doit pousser le pion en position nulle dès que cela est possible, puis viser une nouvelle position nulle à chaque fois que c'est à son tour de jouer : en procédant ainsi, il est sûr d'atteindre le puits. Dans le même temps, son adversaire héritera systématiquement d'une position nulle et ne pourra déplacer le pion que vers une position non nulle : il ne pourra jamais atteindre le puits.

Exemples
Considérons le damier à 5 lignes et 5 colonnes de la figure 3. Le joueur qui commence la partie hérite d'une position nulle et ne peut atteindre qu'une position non nulle. Si le second joueur applique la stratégie gagnante, il est sûr de gagner la partie. En revanche, pour le damier à 5 lignes et 7 colonnes, c'est le joueur qui commence la partie qui est sûr de gagner : il lui suffit de pousser le pion vers une position nulle à chaque fois que c'est à son tour de jouer.

	1	2	3	4	5
1	0	1	3	0	1
2	1	0	2	1	0
3	3	2	0	3	2
4	0	1	3	0	1
5	1	0	2	1	0

	1	2	3	4	5	6	7
1	1	3	0	1	3	0	1
2	0	2	1	0	2	1	0
3	2	0	3	2	0	3	2
4	1	3	0	1	3	0	1
5	0	2	1	0	2	1	0

FIGURE 3 – Celui qui commence perd la partie de gauche et gagne celle de droite

Exercice
Jouer une partie sur un damier à 7 lignes et 10 colonnes contre un adversaire ignorant l'existence d'une stratégie gagnante. L'objectif est de gagner la partie en utilisant la fonction de calcul du nimber pour déterminer le coup gagnant à chaque fois que c'est à vous de jouer.

4 Stratégie gagnante dans le cas général

4.1 La nim-addition

Définition
On définit la **nim-addition** de deux nimbers a et b appartenant à $\{0, 1, 2, 3\}$ de la façon suivante : on convertit a et b en binaire sur 2 bits ; on additionne ensuite les deux codes binaires obtenus sans tenir compte des retenues (*ou exclusif*). Dans tous les cas, le résultat obtenu est le code binaire d'un nimber. Dans la suite, on note $\oplus$ l'opérateur de nim-addition.

Exemples

$$\begin{array}{cccccc} & 1 & \rightarrow & 0 & 1 \\ \oplus & 1 & \rightarrow & 0 & 1 \\ \hline & 0 & \leftarrow & 0 & 0 \end{array} \qquad \begin{array}{cccccc} & 1 & \rightarrow & 0 & 1 \\ \oplus & 2 & \rightarrow & 1 & 0 \\ \hline & 3 & \leftarrow & 1 & 1 \end{array} \qquad \begin{array}{cccccc} & 1 & \rightarrow & 0 & 1 \\ & 2 & \rightarrow & 1 & 0 \\ \oplus & 3 & \rightarrow & 1 & 1 \\ \hline & 0 & \leftarrow & 0 & 0 \end{array}$$

Table de la nim-addition

$\oplus$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Exercice
Quelles sont les propriétés de la nim-addition $\oplus$?

Programmation
Il existe un opérateur du langage C qui réalise exactement la nim-addition. Quel est-il ?

4.2 La stratégie gagnante

Considérons R pions sur un damier. L'état d'une partie est défini par l'ensemble des R cases occupées et de leur nimber. Dans la suite, on note $\{c_1, \ldots, c_k, \ldots, c_R\}$ ces R cases et $\{n_1, \ldots, n_k, \ldots, n_R\}$ leur nimber respectif (i.e. n_k est le nimber de la case c_k).

Position nulle

On pose $p = n_1 \oplus \ldots \oplus n_R$, la nim-addition des nimbers des cases occupées. Lorsque ce nombre p est nul, on dira que les pions sont en position nulle ; dans le cas contraire, on dira que les pions sont en position non nulle. Notons qu'en fin de partie les pions sont en position nulle puisque tous les nimbers sont nuls.

Exemple

Considérons le damier de la figure 4. Les cases occupées sont $(1, 3)$, $(2, 5)$ et $(5, 1)$ de nimbers respectifs 3, 0 et 1. On a $p = 3 \oplus 0 \oplus 1 = 2$ et les pions sont donc en position 2 (non nulle).

Passage d'une position nulle à une position non nulle

Considérons une position nulle $p = n_1 \oplus \cdots \oplus n_k \oplus \cdots \oplus n_R = 0$ et supposons qu'un joueur déplace le pion de la case c_k de nimber n_k vers une voisine c'_k de nimber n'_k. On peut écrire $n'_k = n_k + a$ avec $a \neq 0$ puisque $n'_k \neq n_k$ par construction. La nouvelle position s'écrit alors : $p' = n_1 \oplus \cdots \oplus (n_k \oplus a) \oplus \cdots \oplus n_R \Leftrightarrow p' = p \oplus a \Leftrightarrow p' = a$ puisque $p = 0$. Comme $a \neq 0$, on en déduit que $p' \neq 0$ et la nouvelle position est donc non nulle. Conclusion : si les pions sont en position nulle alors, quelque soit le pion que l'on déplace, on se retrouve systématiquement dans une position non nulle.

Passage d'une position non nulle à une position nulle

Considérons une position $p = n_1 \oplus \cdots \oplus n_k \oplus \cdots \oplus n_R$ non nulle : $p \neq 0$. Pour passer à une position nulle, il suffit de trouver une case c_k de nimber n_k ayant pour voisine une case c'_k de nimber $n'_k = n_k \oplus p$. En effet, si on déplace le pion de c_k vers c'_k, on obtient une nouvelle position de valeur : $p' = n_1 \oplus \cdots \oplus (n_k \oplus p) \oplus \cdots \oplus n_R \Leftrightarrow p' = p \oplus p = 0$.

Par ailleurs, on peut démontrer (on ne le fera pas ici) qu'il existe au moins un coup permettant de passer d'une position non nulle à une position nulle.

Conséquence

La stratégie gagnante est toujours la même : un joueur doit atteindre le plus rapidement possible une position nulle et s'y maintenir jusqu'à la fin de la partie.

Exemple

Les pions, en figure 4, sont en position $p = 3 \oplus 0 \oplus 1 = \mathbf{2}$. On a :

- ☞ La case $(1, 3)$ est de nimber 3 et $3 \oplus \mathbf{2} = \mathbf{1}$. Pour atteindre une position nulle il suffit donc de déplacer le pion de cette case vers une voisine de nimber **1**. D'où le déplacement vers la case $(1, 5)$.
- ☞ La case $(2, 5)$ est de nimber 0 et $0 \oplus \mathbf{2} = \mathbf{2}$. Pour atteindre une position nulle il suffit donc de déplacer le pion de cette case vers une voisine de nimber **2**. D'où le déplacement vers la case $(3, 5)$.
- ☞ La case $(5, 1)$ est de nimber 1 et $1 \oplus \mathbf{2} = \mathbf{3}$. Aucune voisine de cette case ne possède un nimber égal à **3**. Il est donc impossible d'atteindre une position nulle en déplaçant le pion de la case $(5, 1)$.

	1	2	3	4	5
1	0	1	3	0	1
2	1	0	2	1	0
3	3	2	0	3	2
4	0	1	3	0	1
5	1	0	2	1	0

FIGURE 4 – Deux coups gagnants

Exercice
Terminer la partie (figure 4) en utilisant la stratégie gagnante.

5 Quelques consignes

Les paramètres du jeu
On écrira une fonction `Lire_Entier` qui permet de saisir et de retourner un entier compris entre deux bornes données (les valeurs renvoyées doivent être valides). On écrira ensuite une fonction `Lire_Parametres` qui saisira :

- ☞ `N` : le nombre de lignes du damier (compris entre 3 et 30 inclus).
- ☞ `M` : le nombre de colonnes du damier (compris entre 3 et 30 inclus).
- ☞ `R` : le nombre de pions initialement présents sur le damier (compris entre 1 et `N` inclus).
- ☞ `Next` : 1 si c'est à l'ordinateur de jouer ou 2 sinon.
- ☞ `Niveau` : le niveau de difficulté du jeu, 1, 2 ou 3.

Les types
On utilisera les types suivants :

- ☞ `T_Coord` : pour représenter une case par ses coordonnées.
- ☞ `T_Tab_Coord` : un tableau de `T_Coord`.
- ☞ `T_Case` : pour représenter une case par ses coordonnées et son nimber.
- ☞ `T_Tab_Case` : un tableau de `T_Case`.

Représentation de l'état du jeu
L'état du jeu à un instant donné de la partie est défini par la position des pions. Pour représenter cet état, il suffit donc de représenter l'ensemble des cases contenant un pion. On représentera cet ensemble par un tableau de type `T_Tab_Case` (voir illustration en figure 5). Si une case contient $k > 1$ pions, elle sera enregistrée k fois dans la table des cases occupées.

On écrira les fonctions suivantes :

- ☞ `Init_Damier` : une fonction qui construit la table des cases occupées en début de partie.
- ☞ `Contient_Pion` : une fonction qui indique si une case donnée contient ou non un pion.
- ☞ `Affiche_Damier` : une fonction qui affiche le damier à l'écran à partir de la table des cases occupées (il n'est pas nécessaire d'utiliser une table à 2 dimensions).

	1	2	3	4	5
1	0	1	3	0	1
2	1	0	2	1	0
3	3	2	0	3	2
4	0	1	3	0	1
5	1	0	2	1	0

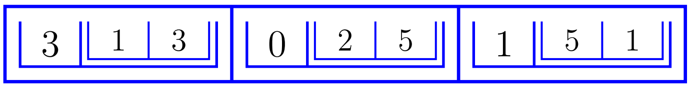

FIGURE 5 – Représentation d'une partie par la table des cases occupées

Le calcul du nimber

On écrira les deux fonctions suivantes :

- `Nimber` : une fonction qui calcule puis retourne le nimber d'une case donnée. Le nimber d'une case peut être calculé à partir de ses deux coordonnées et des nombres `N` et `M`.
- `Position` : une fonction qui calcule et retourne la nim-addition des nimbers des cases occupées.

Les déplacements d'un pion

On écrira les fonctions suivantes :

- `Tab_Voisines` : une fonction qui construit la table des cases voisines d'une case donnée.
- `Hasard` : une fonction qui génère et retourne un nombre au hasard entre 1 et une borne donnée.
- `Modif_Damier` : une fonction qui modifie l'état d'un damier en fonction d'un coup donné.
- `Move_Joueur` : une fonction qui saisit et réalise le coup d'un joueur.
- `Move_Hasard` : une fonction qui réalise un coup choisi au hasard parmi les coups possibles.
- `Move_Gagnant` : une fonction qui réalise un coup gagnant.

6 Un exemple d'exécution

L'affichage du damier est réalisé en mode texte (il n'est pas demandé de gérer des écrans graphiques). Les pions sont représentés par le caractère `O` ci-dessous.

```
Paramètres du jeu
-----------------
nombre de lignes   : 5
nombre de colonnes : 5
nombre de pions    : 3
niveau de 1 à 5    : 3
qui commence ?
l'ordinateur (1) ou le joueur (2) : 2
```

```
C'est parti !
-------------

  1 2 3 4 5
1|O|-|-|-|-|
2|O|-|-|-|-|
3|O|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


A toi de jouer !
choisir un pion 1:(1,1) 2:(2,1) 3:(3,1)
---> 3
choisir la destination 1:(3,2) 2:(3,3) 3:(4,1) 4:(5,1)
---> 2


  1 2 3 4 5
1|O|-|-|-|-|
2|O|-|-|-|-|
3|-|-|O|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


L'ordinateur joue (1,1) ---> (2,1)


  1 2 3 4 5
1|-|-|-|-|-|
2|O|-|-|-|-|
3|-|-|O|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


A toi de jouer !
choisir un pion 1:(2,1) 2:(2,1) 3:(3,3)
---> 3
choisir la destination 1:(3,4) 2:(3,5) 3:(4,3) 4:(5,3)
---> 2


  1 2 3 4 5
1|-|-|-|-|-|
2|O|-|-|-|-|
3|-|-|-|-|O|
4|-|-|-|-|-|
5|-|-|-|-|-|
```

```
L'ordinateur joue : (2,1) ---> (3,1)


  1 2 3 4 5
1|-|-|-|-|-|
2|O|-|-|-|-|
3|O|-|-|-|O|
4|-|-|-|-|-|
5|-|-|-|-|-|


A toi de jouer !
choisir un pion 1:(2,1) 2:(3,1) 3:(3,5)
---> 4
erreur !
---> 3
choisir la destination 1:(4,5) 2:(5,5)
---> 2


  1 2 3 4 5
1|-|-|-|-|-|
2|O|-|-|-|-|
3|O|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


L'ordinateur joue : (2,1) ---> (2,3)


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|O|-|-|
3|O|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


A toi de jouer !
choisir un pion 1:(3,1) 2:(2,3)
---> 1
choisir la destination 1:(3,2) 2:(3,3) 3:(4,1) 4:(5,1)
---> a
erreur !
---> euh
erreur !
---> 4
```

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|O|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|O|-|-|-|-|


L'ordinateur joue : (2,3) ---> (3,3)


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|O|-|-|
4|-|-|-|-|-|
5|O|-|-|-|-|


A toi de jouer !
choisir un pion 1:(3,3) 2:(5,1)
---> 2
choisir la destination 1:(5,2) 2:(5,3)
---> 2


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|O|-|-|
4|-|-|-|-|-|
5|-|-|O|-|-|


L'ordinateur joue : (5,3) ---> (5,5)


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|O|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|


! plus qu'un seul pion en (3,3)

A toi de jouer !
choisir la destination 1:(3,4) 2:(3,5) 3:(4,3) 4:(5,3)
---> 3
```

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|O|-|-|
5|-|-|-|-|-|


L'ordinateur joue : (4,3) ---> (4,4)


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|O|-|
5|-|-|-|-|-|


! plus qu'un seul pion en (4,4)


A toi de jouer !
choisir la destination 1:(4,5) 2:(5,4)
---> 1


  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|O|-|


L'ordinateur joue : (5,4) ---> (5,5)

c'est terminé, tous les pions sont tombés dans le puits

VOUS AVEZ PERDU !
```

7 Modalités

Le projet est à faire en binôme (de préférence étudiants d'un même groupe de TD).

L'évaluation se fera en salle machine lors d'une soutenance de 20 minutes environ par binôme. A l'issue de cette soutenance, vos fichiers sources devront être envoyés par mail à l'enseignant qui vous évalue.

Date de soutenance : jeudi 16 janvier 2014. Un planning des créneaux horaires sera affiché le lundi 13 janvier. N'oubliez pas de vous inscrire.