Un jeu de Nim

1 Description du jeu

Deux joueurs s'affrontent autour d'une grille composée de N lignes et de M colonnes. Au début d'une partie, R pions ($1 \leq R \leq N$) sont positionnés dans la première colonne de la grille, de la case de coordonnées $(1, 1)$ à la case de coordonnées $(1, R)$. En figure 1 (à gauche), on a représenté l'état initial d'une grille lorsque $N = 5$, $M = 5$ et $R = 4$.

À tour de rôle, chaque joueur choisit librement l'un des pions présents sur la grille et le déplace, soit de une ou de deux cases vers la droite, soit de une ou de deux cases vers le bas. Le chevauchement d'un autre pion est autorisé et il est possible que plusieurs pions occupent une même case. La figure 1 (à droite) illustre les différentes possibilités de déplacements d'un pion.

Les déplacements autorisés sont tels que tout pion termine irrémédiablement sa course en case (N, M) appelée **puits** (la case grisée en figure 1). Lorsqu'un pion arrive sur cette case, il ne pourra plus être déplacé et il est éliminé 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 aucun pion sur la grille.

Ce jeu appartient à la famille des jeux de **Nim**. La principale caractéristique de ce type de jeu 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 de la taille de la grille et du nombre initial de pions). Cette stratégie est détaillée dans les paragraphes 3 et 4.

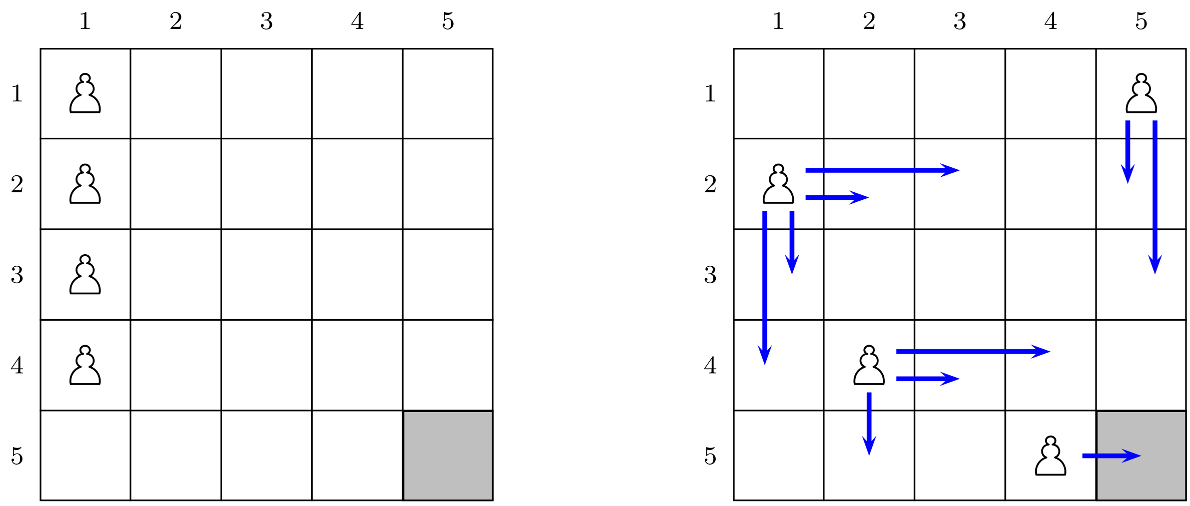

FIGURE 1 – Une configuration initiale et des exemples de déplacements autorisés

2 Travail demandé

Il s'agit d'écrire un programme C permettant de gérer une partie opposant un joueur à l'ordinateur. Les paramètres du jeu seront saisis par le joueur, à savoir : le nombre N de lignes, le nombre M de colonnes, le nombre R de pions initialement présents sur la grille, 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*.

Pour l'ordinateur, jouer au hasard consiste tout d'abord à choisir au hasard un pion parmi les pions présents sur la grille, puis à choisir au hasard un déplacement parmi les déplacements autorisés pour ce pion.

Pour jouer un coup gagnant, l'ordinateur utilisera la stratégie gagnante décrite plus loin. Si aucun coup gagnant n'est possible, il 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
Considérons une partie avec un seul pion sur une grille 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 Les voisines d'une case

On dira qu'une case c' est **voisine** d'une case c, si un pion présent en case c peut être déplacé en case c'. Compte tenu de la règle du jeu, un case possède entre 1 et 4 voisines (voir figure 1) sauf la case (N, M) qui ne possède aucune voisine.

3.2 Le nimber

Considérons une grille et un unique pion en case (1,1). Nous allons associer un nombre entier à chaque case de la grille en utilisant l'algorithme suivant :

(i) Numéroter 0 le puits.

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

(iii) S'il reste des cases non numérotées, 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 la grille. On passe ensuite à l'avant-dernière dernière colonne et on numérote les cases

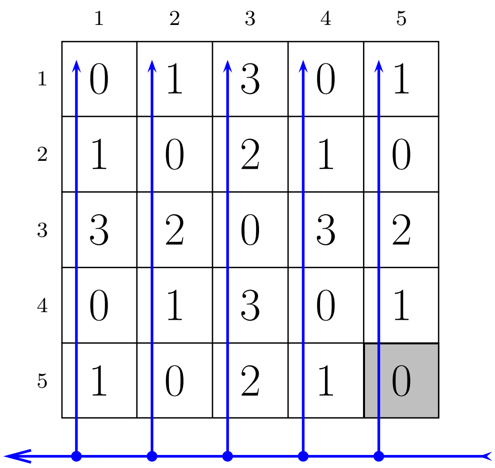

FIGURE 2 – Les nimbers d'une grille à 5 lignes et 5 colonnes

de bas en haut et ainsi de suite, jusqu'à la première colonne de la grille (voir figure 2 ci-dessus). La dernière case numérotée sera donc la case (1,1).

Puisque toute case possède au plus 4 voisines, les 4 entiers 0, 1, 2 et 3, suffisent pour numéroter toutes les cases d'une grille, quelle que soit sa taille. Ces numéros associés aux cases de la grille sont appelés **nim-number** ou **nimber**.

Exercice
Considérons une grille à 7 lignes et 10 colonnes :
- Déterminer le nimber de chaque case.
- Comment calculer le nimber d'une case de cette grille à partir de ses coordonnées, du nombre de lignes et du nombre de colonnes.

Généraliser ensuite en exprimant le nimber d'une case en fonction des coordonnées (i, j) de la case et des dimensions (N, M) de la grille.

3.3 La stratégie gagnante

Définition (La position nulle)
Le but du jeu est d'atteindre le puits dont le nimber est fixé à 0. D'autres cases de la grille ont également 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. Conséquence : **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. Conséquence : **il est toujours possible de déplacer un pion d'une position non nulle vers une position nulle**.

La stratégie
Pour gagner une partie, un joueur doit viser une position nulle à chaque fois qu'il doit jouer. En procédant ainsi, il est sûr d'emmener le pion jusqu'au puits alors que son adversaire, qui héritera systématiquement d'une position nulle, ne pourra que déplacer le pion vers une position non nulle.

Exemple

Considérons la grille à 5 lignes et 5 colonnes ci-dessous. 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 la grille 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 lui 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 une grille 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 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 ?

Remarque
L'opérateur ^ du langage C réalise exactement la nim-addition.

4.2 La stratégie gagnante

Considérons R pions sur une grille. L'état d'une partie est défini par l'ensemble des R cases occupées et de leur nimber. Pour tout $1 \leq k \leq R$, on note n_k le nimber de la case occupée c_k.

Définition (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.*

Exemple
Considérons la grille de la figure 4 page 4. Les cases occupées sont $(1, 3)$, $(2, 5)$ et $(5, 1)$ de nimbers respectifs 3, 0 et 1. La nim-addition est $p = 3 \oplus 0 \oplus 1 = 2$ et les pions sont donc en position 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, quel que 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 3, 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**. La seule possibilité est de déplacer ce pion en 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**. La seule possibilité est de

déplacer ce pion en 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 avec ce pion.

	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 illustrée en figure 4 en utilisant la stratégie gagnante.

5 Quelques consignes

Les paramètres du jeu
On écrira tout d'abord une fonction `Lire_Entier` qui permet de saisir et de retourner un entier compris entre deux bornes données.

On écrira ensuite une fonction `Lire_Parametres` qui saisira :

- ☞ `N` : le nombre de lignes de la grille (compris entre 3 et 30 inclus).
- ☞ `M` : le nombre de colonnes de la grille (compris entre 3 et 30 inclus).
- ☞ `R` : le nombre de pions initialement présents sur la grille (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 :

- ☞ `Coord` : une structure pour représenter une case par ses deux coordonnées.
- ☞ `Case` : une structure pour représenter une case par ses coordonnées et son nimber.

Représentation de l'état du jeu
L'état du jeu à un instant donné est défini par la position des pions sur la grille. Pour représenter cet état, il suffit de stocker les coordonnées des cases contenant un pion dans un tableau de type `Case` (voir figure 5). Si une case contient $k > 1$ pions, elle sera enregistrée k fois dans la table des cases occupées.

Il n'est pas nécessaire d'utiliser une table à 2 dimensions pour représenter une grille.

On écrira les fonctions suivantes :

- `Init_Grille` : 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_Grille` : une fonction qui affiche la grille à l'écran à partir de la table des cases occupées.

	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

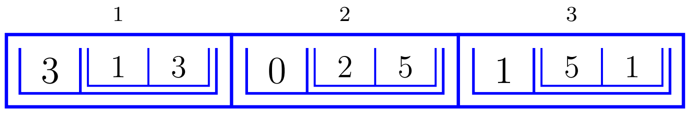

FIGURE 5 – Représentation d'une grille par la table des cases occupées

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 coordonnées (indice de ligne, indice de colonne) et des nombres `N` et `M`.
- `Nim_Addition` : 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 de type `Coord` des cases voisines d'une case donnée.
- `Hasard` : une fonction qui génère et retourne un nombre au hasard entre 1 et une borne donnée.
- `Maj_Grille` : une fonction qui modifie l'état dune grille en fonction d'un coup donné.
- `Move_Joueur` : une fonction qui saisit et réalise le coup d'un joueur (les déplacements autorisés seront présentés sous la forme d'un menu).
- `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 de la grille est réalisé en mode texte (il n'est pas demandé de gérer des écrans graphiques). Les pions sont représentés par le caractère `O`.

```
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, TU AS PERDU !
```

7 Modalités

Le projet est à faire en trinôme.

L'évaluation se fera en salle TP sur les ordinateurs du département. Lors de la soutenance, d'une durée d'au plus 30 minutes (questions comprises), la présence du trinôme est obligatoire. À l'issue de cette soutenance, vos fichiers sources devront être envoyés par mail à l'enseignant qui vous évalue.

Date de soutenance : 18/12/2017

Un planning des créneaux horaires de soutenance sera affiché le : 15/11/2017. N'oubliez pas de vous inscrire !