

# Un jeu de Nim

## 1 Description du jeu

Deux joueurs s'affrontent autour d'une grille quadrillée composée de `nlig` lignes et de `ncol`. En début de partie, un pion est placé dans la case de coordonnées (1, 1). La figure 1 (à gauche) montre l'état de la grille en début de partie lorsque `nlig`=5 et `ncol`=5.

À tour de rôle, chaque joueur doit déplacer le pion, soit de une ou de deux cases vers la droite, soit de une ou de deux cases vers le bas (voir illustration en figure 1, à droite). Les déplacements autorisés sont tels que le pion termine irrémédiablement sa course en case (`nlig`, `ncol`) appelée **puits** (la case *bleue* en figure 1) où il se retrouve bloqué.

**Le gagnant du jeu est celui qui fera tomber le pion dans le puits.**

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. Cette stratégie est détaillée dans le paragraphe 3.

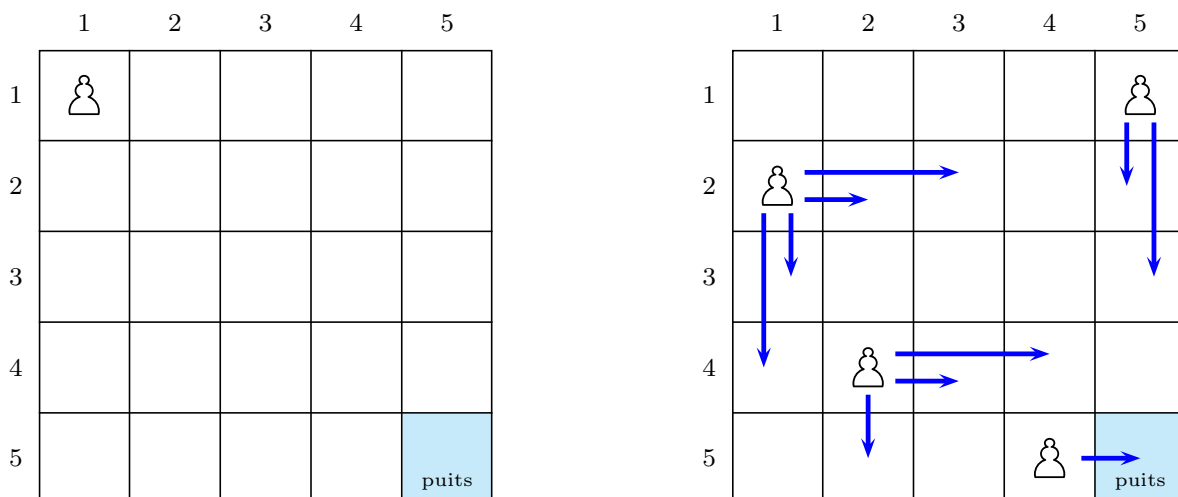


FIGURE 1 – Une configuration initiale et des exemples de déplacements autorisés

Il s'agit d'écrire un programme C permettant de gérer une partie opposant un joueur à l'ordinateur. Ce dernier pourra utiliser la stratégie gagnante. 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.

## 2 Les paramètres du jeu

Les paramètres du jeu seront saisis par le joueur, à savoir : le nombre **nlig** de lignes (compris entre 5 et 30 inclus, par exemple), le nombre **ncol** de colonnes (compris entre 5 et 30 inclus), 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 (4) :

- (1) **niveau débutant** : l'ordinateur joue toujours au hasard.
- (2) **niveau moyen** : l'ordinateur joue un coup au hasard avec la probabilité  $2/3$  ou un coup gagnant avec la probabilité  $1/3$ .
- (3) **niveau expert** : l'ordinateur joue un coup au hasard avec la probabilité  $1/3$  ou un coup gagnant avec la probabilité  $2/3$ .
- (4) **niveau virtuose** : l'ordinateur joue toujours gagnant.

Pour l'ordinateur, jouer au hasard consiste à choisir un coup au hasard parmi les coups possibles. Il est donc possible que le coup choisi au hasard soit un coup gagnant.

Pour jouer un coup gagnant, l'ordinateur utilisera la stratégie gagnante décrite au paragraphe 3. Si aucun coup gagnant n'est possible, il jouera alors un coup au hasard.

## 3 Stratégie gagnante

### 3.1 Les voisines d'une case

On dira qu'une case  $c'$  est **voisine** d'une case  $c$  si un pion placé en case  $c$  peut être déplacé en case  $c'$ . Compte tenu des règles du jeu, toute case autre que le puits possède entre 1 et 4 voisines.

### 3.2 Le nimber

La stratégie gagnante repose sur une numérotation des cases de la grille. A chaque case de la grille, nous allons associer un entier valant soit 0 soit 1. Cet entier est appelé **nimber**, contraction de **nim** **number**. Les nimbers sont déterminés à l'aide de l'algorithme suivant :

- (i) Le nimber du puits est fixé à 0.
- (ii) Tant qu'il existe des cases non numérotées, en choisir une dont toutes les voisines sont numérotées. Soit  $c$  la case ainsi choisie. Si toutes les voisines de  $c$  ont un nimber égal à 1, on attribue le nimber 0 à la case  $c$ , sinon on lui attribue le nimber 1.

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

FIGURE 2 – Ordre de calcul des nimbers d'une grille

A propos de la boucle (ii) : on commence par déterminer le nimber des cases de la dernière colonne, de bas en haut ; on passe ensuite à l'avant-dernière dernière colonne en procédant de la même manière 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).

#### Exercice

Considérons une grille à 7 lignes, 10 colonnes. Déterminer le nimber de chaque case de cette grille en utilisant l'algorithme ci-dessus.

Regarder attentivement les nimbers des cases de cette grille.

En déduire une fonction **Nimber** qui calcule le nimber d'une case de coordonnées (i,j) pour une grille de **nlig** lignes et de **ncol** colonnes.

#### Conclusion

Vous disposez de 2 méthodes pour déterminer les nimbers : soit vous utilisez l'algorithme pour calculer tous les nimbers et les stocker dans une table, soit vous utilisez la fonction **Nimber** pour calculer le nimber d'une case donnée. Il est bien sûr préférable d'utiliser la fonction.

### 3.3 La stratégie gagnante

#### Définition (La position nulle)

On dira qu'un joueur est en **position nulle** si le pion qu'il doit déplacer se trouve dans une case de nimber 0. Si le pion occupe une case de nimber 1, on dira que le joueur qui doit déplacer ce pion est en **position non nulle**. Rappelons que le puits est une case de nimber 0 et correspond donc à une position nulle.

#### Conséquences de la numérotation

- (1) Toute voisine d'une case de nimber 0 est une case de nimber 1. **Quoiqu'il fasse, un joueur en position nulle ne pourra jamais atteindre une autre position nulle mais uniquement une position non nulle.**
- (2) Toute case de nimber 1 possède au moins une voisine de nimber 0. **Il est toujours possible pour un joueur en position non nulle d'atteindre une position nulle.**

#### La stratégie

Pour gagner une partie, un joueur doit donc **viser une position nulle dès que possible et s'y maintenir jusqu'à la fin de la partie**. Son adversaire héritera systématiquement d'une position nulle et ne pourra atteindre qu'une position non nulle : il ne pourra donc jamais atteindre le puits.

#### Exemple

Considérons la grille à 5 lignes et 5 colonnes (figure 3, à gauche). Le joueur qui commence la partie hérite d'une position 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 (figure 3, à droite), c'est le joueur qui commence la partie qui est sûr de gagner car le pion est initialement en position non nulle.

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

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

FIGURE 3 – Pour celui qui commence : position perdante à gauche, position gagnante à droite

#### Exercice

Jouer une partie sur une grille à 7 lignes et 10 colonnes contre un adversaire ignorant l'existence d'une stratégie gagnante. Votre objectif est de gagner la partie en utilisant la stratégie gagnante.

#### Question

Considérons le cas particulier d'une grille ne contenant qu'une seule ligne et un certain nombre de colonnes. Le jeu proposé dans ce TP est alors un jeu bien connu. De quel jeu s'agit-il ?

## 4 Quelques consignes

On note `VMIN` la constante égale au nombre minimum de lignes de la grille (par exemple 5) et `VMAX` la constante égale au nombre maximum de colonnes de la grille (par exemple 30).

### 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 `Parametres` qui saisira :

- ☞ `nlig` : le nombre de lignes de la grille (compris entre `VMIN` et `VMAX` inclus).
- ☞ `ncol` : le nombre de colonnes de la grille (compris entre `VMIN` et `VMAX` inclus).
- ☞ `niveau` : le niveau de difficulté du jeu, 1, 2, 3 ou 4.
- ☞ `next` : 1 si c'est l'ordinateur qui commence, 2 si c'est le joueur.

### Les types

On utilisera les types suivants :

- ☞ `T_Case` : une structure qui représente une case de la grille (deux coordonnées : numéro de ligne, numéro de colonne).
- ☞ `T_Tab_Case` : un tableau à 1 dimension de `T_Case`. Un tableau de ce type permettra de stocker l'ensemble des voisines d'une case donnée.

### Le pion

On note `pion` la variable de type `T_Case` contenant les coordonnées du pion.

### Affichage de la grille

L'état du jeu à un instant donné dépend uniquement de la position du pion. **Il n'est pas nécessaire d'utiliser une table à 2 dimensions pour représenter une grille.** La grille sera affichée après chaque coup. L'affichage sera réalisé en mode texte (affichage graphique non exigé). Le pion sera, par exemple, représenté par le caractère `O`.

### Principales fonctions à écrire

- ☞ `Nimber` : une fonction qui calcule le nimber d'une case donnée.
- ☞ `Coup_joueur` : une fonction qui gère le coup d'un joueur (cette fonction propose au joueur de choisir un coup parmi la liste des coups possibles).
- ☞ `Coup_Ordi_Hasard` : une fonction qui gère le coup au hasard réalisé par l'ordinateur (choisir au hasard une voisine de la case contenant le pion).
- ☞ `Coup_Ordi_Gagnant` : une fonction qui gère le coup gagnant effectué par l'ordinateur (jouer une position gagnante si possible, sinon jouer un coup au hasard).

### Autres fonctions

- ☞ `Voisines` : une fonction qui construit la table des cases voisines d'une case donnée. Cette fonction calcule le nombre de voisines noté `nb_vois` et stocke leurs coordonnées dans une table `vois` de type `T_Tab_Case`.
- ☞ `Hasard` : une fonction qui retourne un nombre au hasard compris entre deux valeurs données.
- ☞ Ainsi que toute autre fonction qu'il vous semblera utile d'écrire.

## 5 Un exemple d'exécution

Paramètres du jeu :

nombre de lignes : 5

nombre de colonnes : 5

niveau de 1 à 4 : 2

qui commence ?

l'ordinateur (1) ou le joueur (2) : 2

C'est parti !

```

  1 2 3 4 5
1|0|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|

```

A toi de jouer !

choisir la destination 1:(1,2) 2:(1,3) 3:(2,1) 4:(3,1)

---> 3

```

  1 2 3 4 5
1|-|-|-|-|-|
2|0|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|

```

L'ordinateur déplace le pion en (2,3)

```

  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|0|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|

```

A toi de jouer !

choisir la destination 1:(2,4) 2:(2,5) 3:(3,3) 4:(4,3)

---> 3

```

  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|0|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|

```

L'ordinateur déplace le pion en (3,4)

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|0|-|-|
4|-|-|-|-|-|
5|-|-|-|-|-|
```

A toi de jouer !

choisir la destination 1:(3,5) 2:(4,4) 3:(5,4)

---> a

erreur !

---> 4

erreur !

---> 2

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|0|-|-|
5|-|-|-|-|-|
```

L'ordinateur déplace le pion en (4,5)

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|-|0|-|
5|-|-|-|-|-|
```

A toi de jouer !

Seule possibilité : (5,5)

valider --->

```
  1 2 3 4 5
1|-|-|-|-|-|
2|-|-|-|-|-|
3|-|-|-|-|-|
4|-|-|-|-|-|
5|-|-|-|-|0|-|
```

C'est terminé. BRAVO TU AS GAGNE !

## 6 Modalités

Ce projet est à réaliser par binôme **du même groupe de TP**. On attend de vous que le programme soit **judicieusement structuré** et **clairement commenté**.

La soutenance se déroulera lors de la dernière séance de TP ([la semaine du 06/01/2025](#)). Lors de cette soutenance, votre fichier source sera **compilé** puis **testé** et une série de **questions** vous seront posées. A la fin de la soutenance, **vous devrez envoyer votre fichier source par mail à l'enseignant**.

## 7 Extras

Le jeu pourra être agrémenté de couleurs, de sons, d'emojis.