

Un jeu de Nim

1 Description du jeu

Deux joueurs s'affrontent autour d'une grille quadrillée composée de n_{lig} lignes et de n_{col} colonnes dont n_{ban} cases sont **bannies**. Au début d'une partie, un pion est placé dans la case de coordonnées (1,1). En figure 1 (à gauche), on a représenté l'état d'une grille en début de partie lorsque $n_{\text{lig}}=5$, $n_{\text{col}}=5$ et $n_{\text{ban}}=2$ (les cases bannies sont *grisées*).

À tour de rôle, chaque joueur déplace le pion, soit de une ou de deux cases vers la droite, soit de une ou de deux cases vers le bas. Les cases bannies sont infranchissables (le chevauchement de ces cases est donc interdit). En figure 1 (à droite), on peut voir quels sont les coups possibles pour diverses positions du pion sur la grille.

Les déplacements autorisés sont tels que le pion termine irrémédiablement sa course en case (n_{lig} , n_{col}) appelée **puits** (la case *bleutée* en figure 1) où il se retrouve bloqué. On dira alors que le pion est tombé dans le puits.

Le gagnant du jeu est celui qui aura fait 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 (cela dépend de la taille de la grille et du nombre de cases bannies). Cette stratégie est détaillée dans le paragraphe 3.

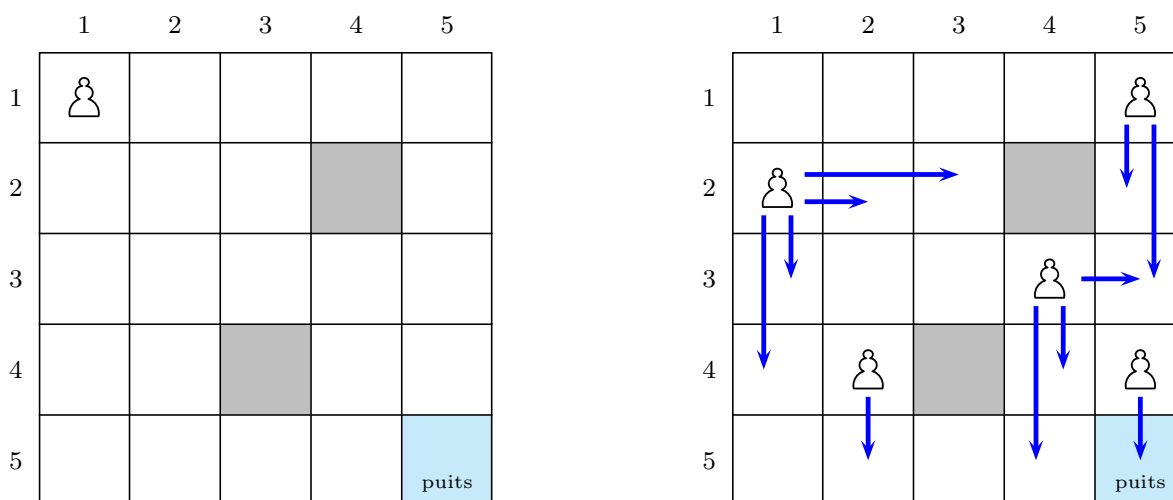


FIGURE 1 – Une configuration initiale et des exemples de déplacements autorisés

2 Travail demandé

Il s'agit d'écrire un programme C permettant de gérer une partie opposant un joueur à l'ordinateur. 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.

2.1 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 $1/3$ ou un coup gagnant avec la probabilité $2/3$.
- (3) **niveau expert** : l'ordinateur joue un coup au hasard avec la probabilité $2/3$ ou un coup gagnant avec la probabilité $1/3$.
- (4) **niveau virtuose** : l'ordinateur joue, si cela est possible, toujours gagnant.

Pour l'ordinateur, jouer au hasard consiste à choisir un coup au hasard parmi les coups possibles.

Attention ! 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.

2.2 Les cases bannies de la grille

Le nombre `nban` de cases bannies sera déterminé aléatoirement (entre 0 et `max(nlig, ncol)` inclus). Puis les cases bannies seront également choisies aléatoirement mais en respectant les contraintes suivantes :

- ☞ La case de départ (1,1) et le puits (`nlig`,`ncol`) ne pourront pas être bannies.
- ☞ Toute case non bannie doit être accessible depuis la case de départ.
- ☞ Le pion ne doit pas se retrouver bloqué dans une case autre que le puits.

Les grilles de la figure 2 contiennent des cases bannies ne respectant pas les contraintes énoncées ci-dessus. Elles ne sont donc pas valides.

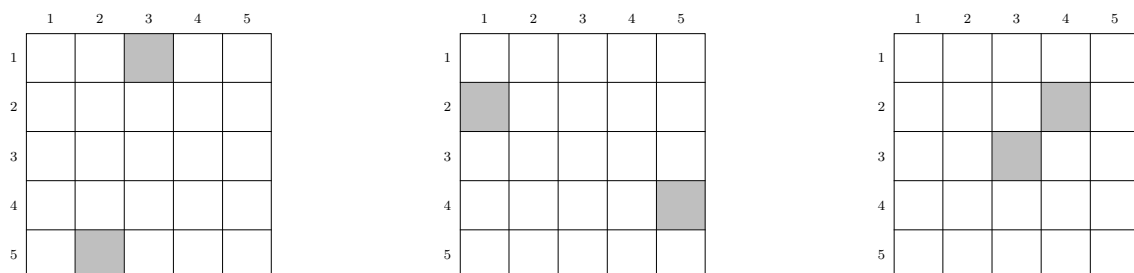


FIGURE 2 – Grilles non valides (les cases bannies ne respectent pas les contraintes)

Exercice

Quelles sont les cases se trouvant en bordure de grille et pouvant être bannies ? Considérons une case bannie valide. Quelle est alors la condition à respecter par les autres cases bannies ?

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 placé en case c peut être déplacé en case c' . Compte tenu des règles du jeu, toute case différente du puits possède entre 1 et 4 voisines.

3.2 Le nimber

La stratégie gagnante repose sur une numérotation des cases non bannies de la grille. Le numéro associé à une grille est un entier valant soit 0 soit 1, appelé **nimber**, contraction de **nim number**. Ces 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 sélectionner une ayant toutes ses voisines numérotées. Soit c la case ainsi sélectionnée. Si toutes les voisines de c ont un nimber égal à 1, on affecte à c le nimber 0, sinon on affecte à c le nimber 1.

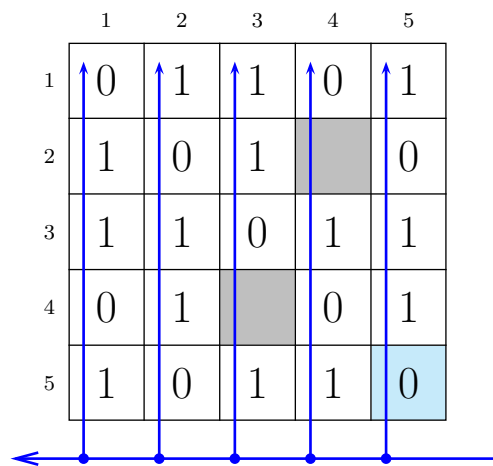


FIGURE 3 – Ordre de calcul des nimbers d'une grille

A chaque itération de la boucle (ii), une case est numérotée. D'un point de vue pratique, 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 3 ci-dessus). La dernière case numérotée sera donc la case (1,1).

Exercice

Considérons une grille à 7 lignes, 10 colonnes et 8 cases bannies (que vous choisirez comme bon vous semble, en respectant les contraintes). Déterminer le nimber de chaque case (non bannie) de cette 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 1.

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

FIGURE 4 – Position perdante à gauche, position gagnante à droite

Conséquences de la numérotation

- (1) Toute voisine d'une case de nimber 0 possède un nimber égal à 1. **Un pion en position nulle ne pourra être déplacé que vers une position non nulle.**
- (2) Toute case de nimber 1 possède au moins une voisine de nimber 0. **Il sera toujours possible de déplacer un pion d'une position non nulle vers 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.**

En procédant ainsi, le joueur 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 et n'atteindra donc jamais le puits.

Exemple

Considérons la grille à 5 lignes et 5 colonnes (figure 4, à gauche). 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 (figure 4, à droite), 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.

Exercice

Jouer une partie sur une grille à 7 lignes, 10 colonnes et 6 cases bannies 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, un certain nombre de colonnes et aucune case bannie. 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.

Cette fonction devra également générer aléatoirement le nombre `nban` de cases bannies (compris entre 0 et `max(nlig,ncol)`).

Les types

On utilisera les types suivants :

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

Le pion

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

Cases Bannies

Écrire une fonction `Hasard_Ban` qui détermine au hasard `nban` cases bannies et les stocke dans une table `ban` de type `T_Tab_Case`. Bien sûr, cette fonction devra tenir compte des contraintes énoncées au paragraphe 2.2.

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.**

Pour l'affichage de la grille (elle doit être affichée après chaque coup), il suffit de connaître la position du pion (variable `pion`) et les positions des `nban` cases bannies (table `ban`). L'affichage sera réalisé en mode texte (il n'est pas demandé d'utiliser le mode d'affichage graphique). Le pion sera, par exemple, représenté par le caractère `O`, et les cases bannies seront signalées par le caractère `X`.

Représentation des nimbers

On utilisera une table d'entiers à deux dimensions : `int nim[VMAX][VMAX]`.

Principales fonctions à écrire

- ☞ **Calcul_Nimbers** : une fonction qui remplit la table **nim** avant le début de la partie.
- ☞ **Coup_joueur** : une fonction qui gère le coup d'un joueur (choix du coup à jouer 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

nombre de cases bannies generees : 2

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|-|-|-|X|-|
3|-|-|-|-|-|
4|-|-|X|-|-|
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|-|-|X|-|
3|-|-|-|-|-|
4|-|X|-|-|-|
5|-|-|-|-|-|

```

L'ordinateur deplace le pion en (2,3)

```

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

```

A toi de jouer !

Seule possibilité : (3,3)

valider --->

```

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

```

L'ordinateur deplace le pion en (3,4)

```

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

```

L'ordinateur deplace le pion en (4,5)

```

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

```

A toi de jouer !

Seule possibilité : (5,5)

valider --->

```

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

```

C'est terminé. BRAVO TU AS GAGNE !

6 Modalités

Ce projet est à réaliser par groupe de **3 étudiants de même groupe de TD**. On attend de vous que le programme soit **judicieusement structuré** et **clairement commenté**. La soutenance se déroulera sur **les ordinateurs des salles machines du Département Informatique**. 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**.

DATE DE SOUTENANCE : 16/01/2020

UN PLANNING DES CRÉNEAUX HORAIRES DE SOUTENANCE SERA AFFICHÉ LE : 13/01/2020. N'OUBLIEZ PAS DE VOUS INSCRIRE !