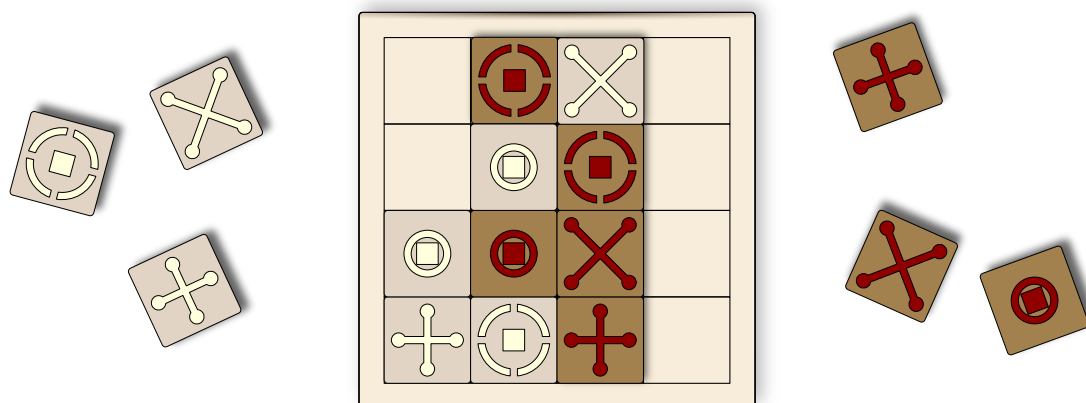


# Bases de la Programmation

## Projet – Mijnlieff

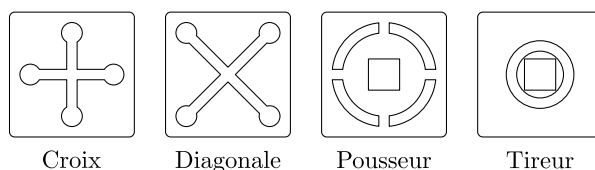
V. Padovani, PPS - IRIF



## 1 Mijnlieff

Mijnlieff est un jeu de damier à deux joueurs, Blanc et Rouge. Il se joue sur un damier carré de taille  $N \times N$  avec  $N \geq 3$ . Dans la version commerciale du jeu,  $N = 4$ , mais le jeu est évidemment jouable sur un damier de taille quelconque.

En début de partie, chacun des joueurs reçoit un ensemble de tuiles de sa couleur. Les joueurs doivent poser à tour de rôle une tuile de leur choix sur une case vide du damier, initialement vide.



Croix

Diagonale

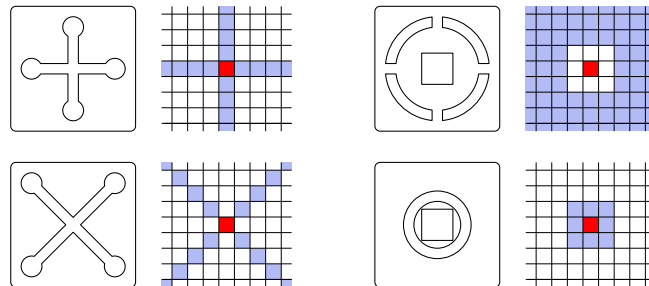
Pousseur

Tireur

**Règles du jeu.** Il y a quatre sortes de tuiles, Croix, Diagonale, Pousseur et Tireur. Le premier à jouer peut poser une tuile sur n'importe quelle case du damier. Après ce premier coup, chaque joueur doit à son tour respecter les contraintes suivantes :

1. Si la dernière tuile posée est une Croix, il ne peut poser une tuile que sur l'une des cases atteintes en se déplaçant horizontalement ou verticalement à partir de cette dernière tuile.
2. Si la dernière tuile posée est une Diagonale, il ne peut poser une tuile que sur l'une des cases atteintes en se déplaçant diagonalement à partir de cette dernière tuile.
3. Si la dernière tuile posée est un Pousseur, le joueur ne peut poser de tuile sur aucune des cases en contact par bord ou par coin avec cette dernière tuile.
4. Enfin, si la dernière tuile posée est un Tireur, le joueur ne peut poser une tuile que sur l'une des cases en contact par bord ou par coin avec cette dernière tuile.

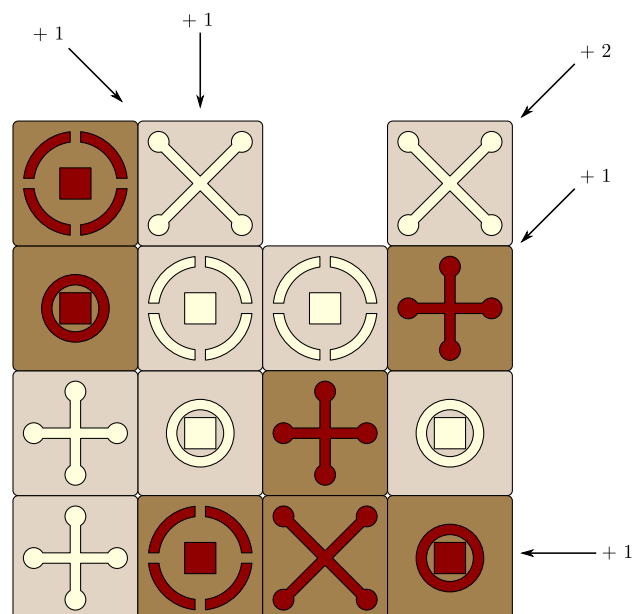
Si le joueur dont c'est le tour ne peut respecter aucune de ces règles, il est contraint de passer son tour. Le joueur opposé est dans ce cas libre de poser la tuile suivante sur une case vide quelconque. Les diagrammes ci-dessous illustrent visuellement ces règles :



**Fin de partie et décompte des scores.** Lorsque l'un des deux joueurs pose sa toute dernière tuile sur le damier, le joueur opposé peut alors, si cela est conforme au règle, poser une toute dernière tuile sur le damier. Après ce dernier coup éventuel, le score de chaque joueur est calculé de la manière suivante :

Chaque joueur compte un point pour chaque rangée de tuiles adjacentes horizontale, verticale ou diagonale de longueur 3. Les rangées de tailles supérieures rapportent 1 point de plus pour chaque tuile supplémentaire (2 points pour 4 tuiles, 3 points pour 5 tuiles, etc.).

Dans la figure ci-dessous, Blanc et Rouge gagnent chacun un point pour une rangée verticale à 3 éléments de leur couleur, et un point chacun pour une rangée diagonale à 3 éléments. Blanc gagne 2 points supplémentaires avec une rangée diagonale de longueur 4. Il remporte donc la partie.



**Distribution initiale du matériel.** Le nombre de tuiles reçues par les joueurs en début de partie dépend de la taille du damier. Chaque joueur reçoit en début partie  $n$  tuiles de chaque sorte, soit  $4 \times n$  tuiles pour chaque joueur et  $8 \times n$  tuiles en tout. La valeur de  $n$  doit être la plus grande valeur telle que  $8 \times n \leq N \times N$ , où  $N$  est la taille du côté du damier en nombre de cases. Par exemple  $n$  vaut 1 pour  $N = 3$ , ou encore  $n$  vaut 3 pour  $N = 5$ , ou encore  $n$  vaut 12 pour  $N = 10$ , etc.

## 2 Projet minimal

Le travail qu'il vous est demandé de réaliser est une implémentation du jeu Mijnlief en langage C. Votre programme devra au minimum implémenter les fonctionnalités suivantes :

1. Il devra permettre de jouer à deux joueurs, chaque joueur proposant à tour de rôle un coup à la machine.

Le damier de jeu devra être affiché en ASCII dans le terminal avec sur son bord des numéros de lignes et de colonnes, et réaffiché après chaque coup joué. L'affichage doit également préciser, pour chaque joueur, combien il reste de tuiles de chaque sorte.

Le programme devra bien sûr vérifier que chaque entrée d'un joueur est correcte, et conforme aux règles du jeu. En cas d'entrée incorrecte, il devra préciser au joueur la nature de son erreur, et lui redemander de spécifier son coup.

Il devra également gérer la fin de la partie, calculer les scores des joueurs et annoncer soit une égalité, soit quel est le joueur gagnant.

2. Il devra permettre de jouer contre l'ordinateur, ce dernier jouant au moins avec une stratégie simpliste (*e.g.* choix aléatoire parmi tous les coups possibles), voire plus élaborée, mais toujours conforme aux règles du jeu.

3. Idéalement, la taille  $N$  du côté du damier devrait pouvoir être choisie librement en début de partie (*c.f.* la Section 2 du Chapitre 7, assez technique). À défaut, elle pourra être définie comme une constante symbolique au début du code (`#define`), avec une contrainte forte : il suffira de modifier cette unique définition de constante dans le code source pour choisir une nouvelle taille de damier, le programme se recompilant et s'exécutant à nouveau sans aucune erreur.

## 3 Critères d'évaluation

Votre programme devra être raisonnablement bien conçu (*cf.* le Chapitre 6 du cours sur les structures) et bien écrit : pas de répétitions dans le code, pas de lignes de plus de 80 caractères ou de fonctions trop longues, choix pertinents de noms de variables et de fonctions, des commentaires sans en faire trop.

Afin de consolider votre note, de nombreuses extensions et variantes du jeu sont envisageables. Toute extension approuvée par votre enseignant sera la bienvenue. Les extensions sont cependant moins importantes que le fait de compléter la partie minimale du projet.

## 4 Organisation du projet

Le projet est à faire de préférence en binôme, éventuellement en monôme – aucun projet à plus de deux ne sera accepté. Veillez à équilibrer les tâches : la simple écriture de la communication avec l'utilisateur (`printf/scanf`) ne vous apprendra rien ; la gestion des règles du jeu est la seule partie du projet qui puisse vous faire avancer en programmation.

Votre code doit évidemment être strictement personnel : il ne doit s'inspirer ni de code trouvé sur le web, ni de celui d'un autre groupe, ni d'une quelconque "aide" trouvée sur un forum. Il doit être compilable tel quel sous Linux. Votre rendu consistera en :

- un fichier `README`, décrivant brièvement comment compiler votre code,
- votre code-source, tout autre fichier utile pour la compilation ou l'exécution,
- un rapport de quelques pages en pdf décrivant vos choix de conception et d'implémentation, les extensions réalisées, les algorithmes non triviaux utilisés, les problèmes non résolus, tout autre élément non immédiat, etc.

Ces éléments, répartis dans un ou plusieurs répertoires, devront être envoyés par email sous la forme d'une archive `tar.gz` portant obligatoirement le nom `nom.tar.gz` si vous êtes en monôme ou `nom_1-nom_2.tar.gz` si vous êtes en binôme. Cette archive devra se décompresser en un répertoire portant le nom `nom` ou `nom_1-nom_2`. Les dates de rendu et de soutenance seront précisées ultérieurement sur la page du cours.

## 5 Remarques

On peut afficher des caractères en couleur dans le terminal.

```
#include <stdio.h>

#define COLOR_RED    "\x1B[31m"
#define COLOR_RESET "\x1B[0m"

int main () {
    printf (" blanc");
    printf (COLOR_RED);    // passage au rouge
    printf (" rouge");
    printf (COLOR_RESET);  // retour a l'affichage normal.
    printf (" blanc\n");
    return 0;
}
```

```
blanc rouge blanc
```