



Initiation à la Programmation C

Devoir maison - L2

Incognito



Nous nous proposons d'implémenter en C un jeu à deux joueurs à information incomplète, c'est-à-dire où chaque joueur ignore certains éléments de la position de jeu.

1 Le jeu Incognito

1.1 Le plateau de jeu

Le plateau de jeu est une grille carrée de taille $T \times T$, où T est un entier défini en tête de fichier par :

```
# define T 5
```

Dans deux coins diamétralement opposés du plateau se trouvent les châteaux respectifs de chaque joueurs (par exemple, les cases $a5$ et $e1$; voir Fig. 1). Chaque joueur possède N pions de sa couleur, sachant que N est calculé comme suit :

$$N = \frac{(T-1)(T-2)}{2} - 1. \quad (1)$$

Dans la pratique, les plateaux de jeu disponibles vérifient $T \leq 8$.

1.2 Chevaliers et espion

Chaque joueur possède un pion (et un seul) marqué sur sa face cachée d'un signe d'allégeance à son château. Celui-ci est alors désigné par le terme d'*espion*.

Les $N - 1$ autres pions du joueur sont des *chevaliers* ayant prêté allégeance au même seigneur.

1.3 But du jeu

Le but du jeu est de faire parvenir son espion dans le château adverse ou de découvrir l'espion ennemi.

1.4 Position de départ

Le plateau de jeu contient au début du jeu N pions noirs et N pions blancs placés dans les angles opposés du plateau et protégeant respectivement les châteaux, comme indiqué sur le plateau de la figure 1 :

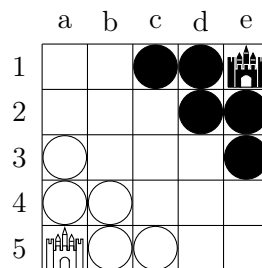


Figure 1: La position initiale du jeu Incognito

Les joueurs ne connaissent pas la position initiale de leur espion, ni celle de l'espion adverse. Pour cette raison, le pion espion de chaque joueur est choisi au hasard parmi ses cinq pions.

1.5 Marche du jeu

Le premier joueur à jouer est choisi au hasard. Chacun à leur tour, les joueurs peuvent choisir entre une (et une seules) des deux sortes de coups :

- *bouger une pièce (et une seule) dans une quelconque des huit directions*, droites ou diagonales, d'autant de cases voulues, sans pouvoir cependant sauter par-dessus une pièce qu'elle soit amie ou ennemie (c'est le déplacement d'une dame aux échecs). Une pièce ne peut pas non plus aller sur une case déjà occupée.
- *interroger une pièce ennemie*. Ceci n'est possible que lorsque deux pièces adverses sont adjacentes orthogonalement (par exemple a2-b2 ou a2-a3, mais pas a2-b1). Le joueur interroge donc la pièce ennemie avec une de ses propres pièces. Plusieurs cas se distinguent alors :
 - si la pièce interrogée est l'espion de l'adversaire, le questionnant gagne ;
 - si la pièce interrogée n'est pas l'espion, le questionnant perd la pièce avec laquelle il a interrogé.

Dans le cas où la pièce perdue était l'espion du joueur, il perd la partie.

Cette interrogation entre deux pièces adjacentes n'est pas obligatoire et ne peut se faire qu'à la place d'avoir déplacé une pièce.

Un joueur ne peut pas faire entrer une de ses pièces dans son propre château. Si un joueur fait parvenir dans le château adverse une pièce qui n'est pas l'espion, celle-ci est perdue. Par ailleurs, on ne peut entrer dans un château qu'en ligne orthogonale (donc par un des deux côtés auxquels appartient le coin).

1.6 Fin de partie

Deux raisons permettent de décider le vainqueur.

1. Un espion a été démasqué :

- ↪ si un espion interroge une pièce adverse qui n'est pas un espion, alors le joueur ayant interrogé a perdu.
- ↪ si une pièce interroge une pièce adverse qui est un espion, alors le joueur ayant interrogé a gagné.

2. Un château a été pris :

- ↪ Un joueur qui arrive à faire rentrer son espion dans le château adverse est déclaré gagnant.

2 Implémentation

Pour implémenter ce jeu, nous définissons tout d'abord une constante :

```
#define TAILLE 5
```

Nous définissons ensuite deux énumérations, utiles plus tard pour définir facilement les pions :

```
typedef enum _couleur{BLANC, NOIR} Couleur ;  
typedef enum _type{CHEVALIER, ESPION} Type ;
```

Une énumération en C (enum) est, comme les structures, un type de données défini par l'utilisateur en C. Une énumération peut être utilisée comme ici, pour attribuer des noms (BLANC, NOIR, CHEVALIER et ESPION) à des constantes de type entières. Ces noms ont pour but de faciliter la lecture du code et sa maintenabilité. Les variables symbolique BLANC, NOIR, CHEVALIER et ESPION sont alors manipulées comme si elles avaient définies avec un `#define`.

Enfin, nous utiliserons les structures suivantes pour modéliser un Pion, une Case, une Direction, un mouvement ainsi que le Jeu :

```
typedef struct _pion {
    Type type;
    Couleur couleur;
} Pion;

typedef struct {
    int x, y;
} Case, Direction;

typedef struct _jeu {
    Pion * plateau[TAILLE][TAILLE];
    Couleur joueur;
} Jeu;

typedef struct _mouvement {
    Case depart;
    Case arrivee;
} Mouvement;
```

3 Sauvegarde d'un déroulé de jeu

Afin de pouvoir interrompre une partie et la reprendre plus tard, ou de pouvoir tester plusieurs stratégies à une étape du jeu, nous avons besoin de sauvegarder tous les coups qui ont été joués dans un fichier et de pouvoir charger une partie commencée.

Pour ce faire, nous allons décrire le format de fichier `.inco` permettant de sauvegarder une partie en cours. Un exemple complet d'un tel fichier est donné en Fig. ??, à l'annexe §5. La figure ?? donne le descriptif complet de la partie d'incognito correspondante.

Tous les pions du plateau de jeu seront repérés sur celui-ci sous la forme d'une lettre suivi d'un chiffre. Par exemple, le château blanc est en `a5` alors que le château noir est en `e1`.

Sur la première ligne d'un fichier `.inco`, on trouve la position initiale de l'espion blanc, indiquée par la lettre `B` suivi d'un espace, puis de la localisation de celui-ci sur le plateau de jeu.

Sur la deuxième ligne, on trouve la position de l'espion noir. La lettre `B` de la première ligne est alors remplacé par un `N`.

Sur la troisième ligne, on trouve une lettre (`B` ou `N`) qui indique le joueur commençant la partie.

Ensuite, chaque coup joué est indiqué sur une nouvelle ligne. Celle-ci commence par une lettre, `D` pour un déplacement et `I` pour une interrogation. Suit alors un espace, puis la localisation d'une pièce, de la flèche `->` et enfin d'une autre localisation de pièce du jeu.

Tous les coups seront enregistrés dans le fichier `.inco`, même les coups non-valides. Lors de la lecture, si une ligne contient un coup qui n'est pas valide, celle-ci sera simplement ignorée.

Pour indiquer la fin d'une partie, la dernière ligne contiendra uniquement la lettre `B` ou `N` indiquant la couleur du joueur gagnant.

Lors de la restauration d'une partie précédemment sauvegardé, l'extension du fichier de sauvegarde devra être vérifié ainsi que son formatage.

4 Consignes

4.1 Réalisation

Le devoir maison est à réaliser **en binôme au sein d'un même groupe de TP**.

4.2 Main

Votre rendu permettra de produire un exécutable dont le comportement dépendra des paramètres transmis sur la ligne de commande. Il acceptera les options :

- -a pour indiquer que l’affichage sera en ASCII (option excluant l’option suivante) ;
- -g pour indiquer que l’affichage sera graphique (option excluant l’option précédente) ;
- -s pour indiquer que chaque coup des joueurs sera sauvegardé dans le fichier dont le chemin suivra l’option dans la ligne de commande
- -c pour indiquer de reprendre une partie sauvegardée, le chemin du fichier de sauvegarde étant donné à la suite de l’option.

Par exemple, la ligne de commande suivante permettra de charger la partie `sauvegarde_1.inco` en mode graphique et de sauvegarder tous les coups joués sur le fichier `sauvegarde_1.1.inco` :

```
./Incognito -g -c sauvegarde_1.inco -s sauvegarde_1.1.inco
```

Remarque : Le fichier `sauvegarde_1.1.inco` commencera comme le fichier `sauvegarde_1.inco`. Les nouveaux coups joués seront ajouté à la suite.

En cas d’absence d’option, vous pourrez choisir le comportement par défaut (par exemple, le mode graphique, couplé à la sauvegarde de chacun des coups joués).

Remarques importantes :

- Il est fortement conseillé de commencé à développer l’affichage en ascii, avant de développer celui en mode graphique !
- L’affichage en ascii devra se conforter scrupuleusement à celui présenté au cours d’un exemple complet de partie à l’annexe §5.

4.3 Documentation

Chaque fonction sera documentée, avec à minima une description de :

1. la fonction ;
2. ses arguments ;
3. sa valeur de retour ;
4. ses éventuels effets de bords, avec éventuellement un exemple.

Un rapport sera à rendre avec votre code. Celui-ci contiendra quatre parties :

- une documentation utilisateur ;
- une documentation technique, pour de futurs développeurs devant reprendre votre travail ;
- la méthode de travail en équipe avec une description ...
 - 1°) ... de la méthodologie de travail mise en œuvre par l’équipe ;
 - 2°) ... des difficultés rencontrées et de leurs résolutions ;
- les potentielles améliorations réalisées.

4.4 Rendu

Le rendu sur e-learning contiendra un et un seul fichier `.zip` dont le nom sera formaté comme suit : `nomUn_nomDeux.zip` Il contiendra deux fichiers, le `.c` et le rapport nommé respectivement selon la convention :

```
nomUn_nomDeux_-_Incognito.c
nomUn_nomDeux_-_Rapport.pdf
```

Enfin, le rendu devra être déposé avant le DIMANCHE 27 OCTOBRE 2024, 23h59.

4.5 Barème

Le barème du devoir maison sera :

Non respect des consignes :		-1, -2, ...
Version ascii :	Création en mémoire du plateau de jeu	1
	Déplacement d'un pion	4
	Interrogation d'un pion	1
	Gestion de la fin de jeu	1,5
Sauvegarde / chargement d'un jeu :	Sauvegarde	3
	Chargement	2
Version graphique :		2
Gestion des options de l'exécutable :		2
Documentation :	Rapport	1,5
	Documentation des fonctions et commentaires du code	2
Améliorations potentielles :	(notation par amélioration)	+0,5 à +1

Enfin, ...

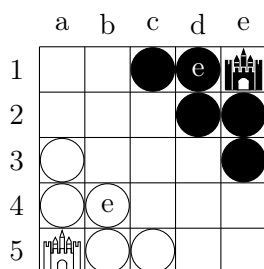
... FAITES VOUS PLAISIR ! MAIS...

Attention : Vos améliorations ne seront prises en compte que si la version ASCII et la version graphique fonctionnent parfaitement.

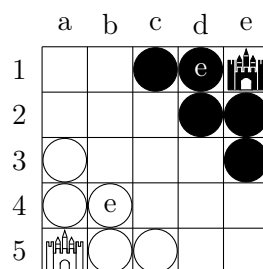
5 Annexes

Voici un exemple complet d'une partie d'incognito, en mode graphique, puis en mode textuel. Enfin, la partie est sauvegardé intégralement et disponible en §5.3.

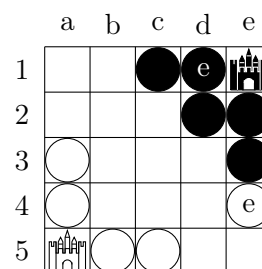
5.1 Une partie complète d'Incognito



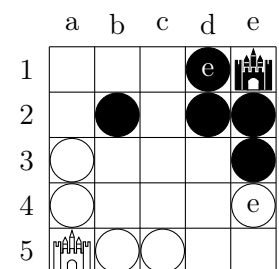
Position initiale



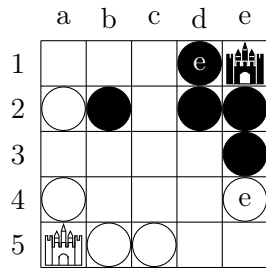
Ligne D b5->a5
Coup non valide



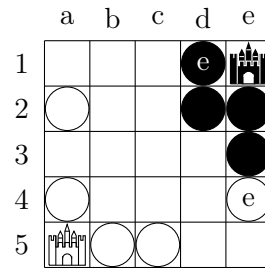
Ligne D b4->e4



Ligne D c1->b2

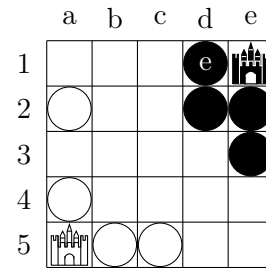


Ligne D a3->a2



Ligne I b2->a2

Le pion b2 est perdu.



Ligne I e4->e3

Le pion e4 est perdu :
fin de la partie...

Les noirs ont gagnés !

5.2 Le déroulé de la même partie en mode textuel

(base) bouillot % ./Incognito

Bienvenu dans Incognito

```

-----
| | |n|n| |
-----
| | | |n|n|
-----
|b| | | |n|
-----
|b|b| | | |
-----
| |b|b| | |
-----

```

Rappel : (0, 0) designe le coin supérieur gauche et (4, 4) le coin inférieur droit

Joueur blanc, voulez vous faire un déplacement ou une interrogation ? ('d' ou 'i')

d

Quel déplacement pour le joueur blanc ?

Saisie sous la forme (a, b) --> (c, d).

(4, 1) --> (4, 0)

Déplacement non licite : un joueur ne peut pas amener un pion sur son chateau.

Quel déplacement pour le joueur blanc ?

Saisie sous la forme (a, b) --> (c, d).

(3, 1) --> (3, 4)

```

-----
| | |n|n| |
-----
| | | |n|n|
-----
|b| | | |n|
-----
|b| | | |b|
-----
| |b|b| | |
-----

```

Joueur blanc, voulez vous faire un déplacement ou une interrogation ? ('d' ou 'i')

d

Quel déplacement pour le joueur noir ?

Saisie sous la forme (a, b) --> (c, d).

(0,2) --> (1, 1)

```
-----
| | | |n| |
-----
| |n| |n|n|
-----
|b| | | |n|
-----
|b| | | |b|
-----
| |b|b| | |
-----
```

Joueur blanc, voulez vous faire un déplacement ou une interrogation ? ('d' ou 'i')

d

Quel déplacement pour le joueur blanc ?

Saisie sous la forme (a, b) --> (c, d).

(2, 0) --> (1, 0)

```
-----
| | | |n| |
-----
|b|n| |n|n|
-----
| | | | |n|
-----
|b| | | |b|
-----
| |b|b| | |
-----
```

Joueur blanc, voulez vous faire un déplacement ou une interrogation ? ('d' ou 'i')

i

Quel pion noir est l'interrogateur ?

Saisie sous la forme (a, b)

(1, 1)

Quel pion est questionné ?

Saisie sous la forme (a, b)

(1, 0)

Joueur noir, vous n'avez pas trouvé l'espion de votre adversaire...

Et en plus, votre chevalier s'est fait empoisonné...

```

-----
| | | |n| |
-----
|b| | |n|n|
-----
| | | | |n|
-----
|b| | | |b|
-----
| |b|b| | |
-----

```

Joueur blanc, voulez vous faire un déplacement ou une interrogation ? ('d' ou 'i')

i

Quel pion blanc est l'interrogeur ?

Saisie sous la forme (a, b)

(3, 4)

Quel pion est questionné ?

Saisie sous la forme (a, b)

(2, 4)

Joueur blanc, vous avez interrogé un chevalier avec votre espion...

Le gagnant est le joueur noir : BRAVO !

5.3 Le fichier de sauvegarde .inco associé à la partie

```

B b4
N d1
B
D b5->a5
D b4->e4
D c1->b2
D a3->a2
I b2->a2
I e4->e3
N

```