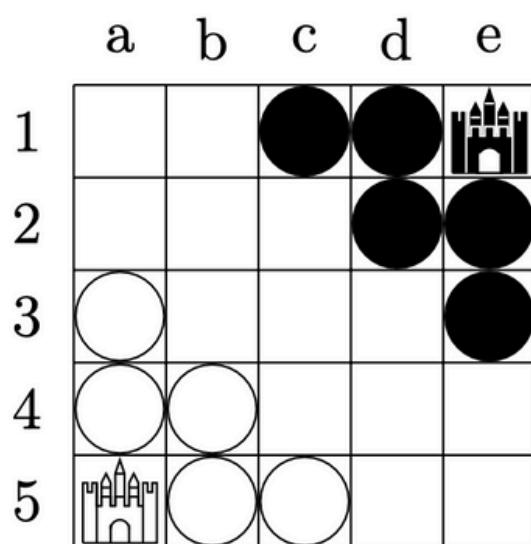


RAPPORT DEVOIR MAISON PROGRAMMATION C

OCTOBRE 2024

INCOGNITO



Préparé par :
Inès Benhamida

TABLE DES MATIÈRES

I. DOCUMENTATION UTILISATEUR	1
II. DOCUMENTATION TECHNIQUE	2
III. MÉTHODE DE TRAVAIL	3
IV. POTENTIELLES AMÉLIORATIONS	3

I. Documentation utilisateur

Description générale

Le projet Incognito est un jeu stratégique de plateau simulant la capture d'espions. Deux joueurs s'affrontent en tentant de faire entrer leur espion dans le château adverse ou de découvrir l'espion adverse. Le jeu se joue en ligne de commande et permet des mouvements ainsi que des interrogations de pions pour identifier l'espion de l'adversaire.

Installation

1. Téléchargez le code source du jeu.
2. Compilez le programme avec la commande suivante :

```
clang -std=c17 -Wall -Wfatal-errors BENHAMIDA_DAABAK_-_Incognito.c -o incognito
```

3. Lancez le programme avec la commande :

```
./incognito -a -s <fichier_de_sauvegarde>.inco
```

- Le paramètre -a désigne l'affichage en ASCII
- Le paramètre -s permet de spécifier le fichier de sauvegarde pour enregistrer ou reprendre une partie.
- Utilisez -c <nom_fichier> pour charger une partie sauvegardée.

Règles du jeu

Objectif : Le joueur doit faire entrer son espion dans le château adverse ou découvrir l'espion adverse.

Déplacements : À chaque tour, le joueur peut déplacer un pion orthogonalement ou diagonalement s'il n'y a aucun obstacle entre le départ et l'arrivée.

Interrogations : Les pions peuvent interroger les pions adjacents orthogonalement pour vérifier s'ils sont des espions. Si un espion est découvert, l'adversaire perd immédiatement la partie.

Conditions de victoire : Un joueur gagne en introduisant son espion dans le château adverse ou en révélant l'espion adverse via une interrogation.

Comment Jouer à “Incognito”

1. Démarrage du jeu

Lancez l'exécutable du jeu à partir de la ligne de commande avec les options souhaitées.

Par exemple :

```
./Incognito -a -s sauvegarde.inco
```

-a : mode ASCII pour l'affichage textuel.

-s : pour sauvegarder les coups dans le fichier sauvegarde.inco

2. Les actions possibles

À chaque tour, un joueur peut choisir entre deux actions :

1. Déplacement d'une pièce :

- Entrez d pour choisir un déplacement.
- Ensuite, entrez les coordonnées de départ et d'arrivée sous la forme (x1, y1) --> (x2, y2).
Par exemple, pour déplacer un pion de la case a2 à la case b3, entrez : **(1, 0) --> (2, 1)**
- Les coordonnées sont basées sur un système où (0, 0) correspond au coin supérieur gauche et (T-1, T-1) au coin inférieur droit.

2. Interrogation d'une pièce :

- Entrez i pour choisir une interrogation.
- Vous devrez indiquer les coordonnées de votre pion interrogateur et celles de la pièce ennemie à interroger. Par exemple, si votre pion est en b2 et que vous interrogez un pion ennemi en b3, entrez : **(1, 1) (2, 1)**

3. Règles importantes

- Vous ne pouvez pas sauter par-dessus d'autres pièces.
- Les déplacements doivent respecter les directions orthogonales ou diagonales, comme une dame aux échecs.
- Une pièce ne peut pas aller sur une case déjà occupée.
- Pour entrer dans le château adverse, utilisez une entrée orthogonale. Si une pièce autre que l'espion y entre, elle est perdue.

4. Fin de partie

- La partie se termine lorsqu'un espion est découvert ou lorsqu'un espion entre dans le château adverse.
- Le vainqueur sera indiqué à l'écran, et le fichier de sauvegarde sera mis à jour si l'option est activée.

5. Messages d'erreur courants

- Déplacement non valide : Un message s'affichera si vous essayez un coup interdit, comme déplacer un pion sur une case occupée ou entrer dans votre propre château.
- Interrogation incorrecte : Si vous interrogez un chevalier ennemi, vous perdrez la pièce interrogatrice.

II. Documentation technique

Structures du code

Types et structures :

- Type : Enumération pour différencier les types de pions (CHEVALIER et ESPION).
- Couleur : Enumération pour les équipes de joueurs (BLANC et NOIR).
- Pion : Structure contenant le type et la couleur d'un pion.
- Jeu : Structure principale contenant le plateau de jeu et le joueur en cours.
- Case : Structure pour représenter les coordonnées d'une case sur le plateau.

Fonctions principales :

- initialiser_plateau(Jeu* jeu) : Initialise le plateau de jeu avec la disposition initiale des pions.
- afficher_plateau(Jeu* jeu) : Affiche l'état actuel du plateau.
- deplacement_valide(Case depart, Case arrivee, Jeu* jeu) : Vérifie si un déplacement est valide selon les règles du jeu.
- deplacer_pion(Case depart, Case arrivee, Jeu* jeu) : Déplace un pion d'une case à une autre si le déplacement est valide.
- interroger_pion(Case interrogateur, Case interroge, Jeu* jeu) : Permet à un pion d'interroger un pion adjacent pour découvrir l'espion adverse.
- sauvegarder_partie(Jeu* jeu, const char* fichier, char type_coup, Case depart, Case arrivee) : Sauvegarde l'état actuel de la partie.
- charger_partie(Jeu* jeu, const char* fichier) : Charge une partie sauvegardée depuis un fichier.

Gestion des erreurs et messages :

- Chaque action de déplacement ou d'interrogation renvoie un code de statut pour indiquer la validité de l'action.
- Des messages d'erreur sont affichés pour guider l'utilisateur dans les cas d'erreurs (déplacement hors limites, entrée non autorisée dans le château, etc.).

Gestion des fichiers de sauvegarde

Les parties sont sauvegardées dans des fichiers .inco. Les fichiers enregistrent :

1. La position initiale des espions pour chaque couleur.
2. Le joueur qui commence.
3. La liste des coups (mouvements et interrogations).
4. L'état final, indiquant le gagnant si la partie est terminée.

III. Méthode de travail en équipe

Pour ce projet, nous avons travaillé ensemble à distance en utilisant des outils de communication en ligne. Nous avons régulièrement partagé nos avancées, discuté des problèmes rencontrés et proposé des solutions. Chaque membre de l'équipe a contribué à des parties spécifiques du code tout en assurant une bonne cohérence globale.

Difficultés rencontrées et solutions

Implémentation des règles de déplacement :

Nous avons rencontré des difficultés pour gérer les règles de déplacement, en particulier pour les restrictions liées aux châteaux. Après plusieurs tests et ajustements, nous avons modifié les fonctions de déplacement pour respecter les règles du jeu.

Gestion des sauvegardes et chargement de parties :

Certaines parties sauvegardées ne conservaient pas l'état correct du plateau. En ajoutant des vérifications et des tests, nous avons réussi à corriger les fonctions de sauvegarde et de chargement pour garantir une reprise correcte des parties.

IV. Potentielles améliorations

Interface graphique : Une interface graphique en complément de la version en ligne de commande rendrait le jeu plus attractif et facile à utiliser pour les utilisateurs.

Optimisation du code : Certaines fonctions, comme `deplacer_pion`, pourraient être optimisées pour une meilleure gestion des erreurs et une exécution plus rapide, notamment en réduisant le nombre de vérifications redondantes.