UNIVERSITE ABDELMALEK ESSAADI FACULTE DES SCIENCES ET TECHNIQUES DE TANGER

PROJET DE FIN DE MODULE
CYCLE GÉO-INFORMATION

Réalisation d'un jeu de dames en Java orienté objet

Réalisé par :
El Khlifi OUMAIMA
Hamdach HAFSSA

Encadré par :
Mayouche IBTIHAL

Année Universitaire 2024-2025

Nous exprimons notre gratitude envers Allah pour les bénédictions qui nous ont permis d'être entourés de personnes aimables et serviables, de jouir d'une bonne santé et d'avoir la force et le courage nécessaires pour mener à bien ce travail.

Au terme de ce travail, nous souhaitons tout d'abord adresser toute notre gratitude à notre professeur, **Mme Mayouche Ibti** ,pour sa confiance, son accompagnement et les précieux conseils qu'elle nous a prodigués tout au long de ce projet.

Ce projet nous a permis d'apprendre énormément et d'acquérir de nouvelles compétences en programmation orientée objet en Java.

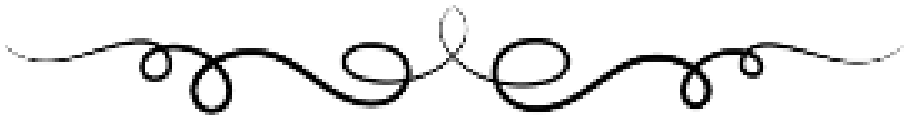

Table des matières

Chapitre 1

Introduction Générale

1.1 Contexte général

Ce projet s'inscrit dans le cadre du module de Programmation Orientée Objet (POO) de notre cursus en informatique. Il consistait à concevoir et développer une application Java simulant le jeu de dames, un jeu de stratégie classique, en respectant rigoureusement les règles officielles.

L'objectif principal était de mettre en œuvre les concepts fondamentaux de la POO, tels que :

— **L'encapsulation**, pour protéger les données des objets et garantir leur intégrité.

— **L'héritage**, permettant de définir des classes génériques pour les pièces du jeu et de spécialiser leurs comportements.

— **Le polymorphisme**, pour adapter les comportements des différentes pièces sur le plateau via des méthodes redéfinies.

— **La gestion des exceptions**, pour traiter de manière robuste les erreurs d'entrées ou les mouvements illégaux.

Le projet a été réalisé en deux phases :

— Une première phase de développement en mode console, permettant de se concentrer sur la logique métier et les règles du jeu.

— Une seconde phase d'évolution avec la création d'une interface graphique en Java Swing, rendant l'application plus conviviale et interactive.

L'utilisation de **Swing**, la bibliothèque graphique standard de Java, a permis de concevoir une interface utilisateur avec un plateau de jeu visuel, des cases cliquables et une gestion intuitive des déplacements des pièces. Cette interface a enrichi l'expérience utilisateur en offrant un rendu visuel clair, des retours immédiats sur les actions, ainsi que des messages d'alerte en cas de mouvements invalides.

Ce projet nous a ainsi permis de combiner efficacement les concepts théoriques de la POO avec la réalisation d'une application graphique complète, tout en développant des compétences pratiques en conception, programmation et gestion d'interfaces utilisateur.

1.2 Principe du Jeu de Dames

Le jeu de dames est un jeu de stratégie qui oppose deux joueurs sur un plateau de 64 cases (8 lignes × 8 colonnes), alternativement noires et blanches. Seules les cases noires sont utilisées pour jouer. Chaque joueur dispose de 12 pions placés sur les trois premières rangées de son côté.

Les règles principales sont les suivantes :

Déplacement

— Les pions se déplacent uniquement en diagonale vers l'avant, d'une seule case.
— Ils ne peuvent pas reculer sauf s'ils sont promus en Dames.

Capture

— Un pion peut capturer une pièce adverse en sautant par-dessus elle, à condition que la case suivante soit vide.
— La capture est obligatoire dès qu'elle est possible.
— Si plusieurs captures sont possibles, un enchaînement de prises doit être réalisé dans le même tour.

Promotion

— Lorsqu'un pion atteint la dernière ligne adverse, il devient une **Dame**.
— La Dame peut se déplacer en diagonale, vers l'avant et vers l'arrière, sur plusieurs cases consécutives, tant que le chemin est libre.

Enchaînement de captures

— Après une première capture, si d'autres captures sont possibles avec la même pièce, elles doivent être réalisées dans le même tour.
— Ce mécanisme stratégique permet des prises multiples successives.

Chapitre 2

Conception et Architecture de l'Application

2.1 Structure des Classes

Le projet est organisé en plusieurs classes, chacune ayant un rôle spécifique :

Board — Gère la logique du plateau (8x8).
- Contient les méthodes pour valider les mouvements, effectuer des prises et promouvoir les pions en dames.
- Utilise deux listes (`redPieces` et `bluePieces`) pour suivre les pièces en jeu.

Piece — Représente une pièce (pion ou dame) avec ses coordonnées (`row`, `col`) et son type (`RED_PIECE`, `BLUE_KING`, etc.).
- Méthodes : `move()`, `promoteToKing()`.

Player — Stocke le nom du joueur et sa couleur (`RED` ou `BLUE`).

CheckersGame (Mode Console) — Boucle principale du jeu.
- Gère les entrées utilisateur et affiche le plateau via `Board.display()`.

CheckersGUI (Mode Graphique) — Utilise `Swing` pour l'affichage.
- Gère les interactions souris et met à jour l'interface dynamiquement.

GameConstants — Centralise les constantes (couleurs, taille du plateau, symboles des pièces).

Main — Point d'entrée : propose à l'utilisateur de choisir entre GUI et console.

Développement de l'Interface Graphique

Composants Swing Utilisés

- **JFrame** : Fenêtre principale.
- **JPanel** : Plateau de jeu (grille 8x8).
- **JButton** : Cases cliquables.
- **JLabel** : Affichage des scores et du temps.

2.1.1 Rendu des Pièces

- Utilisation de `ImageIcon` et `BufferedImage` pour dessiner les pions et les dames.

2.1.2 Gestion des Événements

- **Clic sur une case :**
  - Si aucune pièce n'est sélectionnée, vérifie si la case contient une pièce du joueur actuel.
  - Si une pièce est sélectionnée, valide le mouvement via `isValidMove()`.

2.2 Optimisations et Bonnes Pratiques

Modularité

— Séparation claire entre logique métier (`Board`) et affichage (`CheckersGUI` / `CheckersGame`).
— Utilisation de constantes globales (`GameConstants`) pour éviter les « magic numbers ».

2.2.1 Gestion des Erreurs

— Validation des entrées en console avec `try-catch` pour les coordonnées.
— Affichage de messages d'erreur dans l'interface graphique via `JOptionPane`.

2.2.2 Performances

— **Timer** : Utilisation de `java.util.Timer` pour le chronomètre.
— **Rendu graphique :**
  — Double buffering implicite avec Swing.
  — Utilisation de `revalidate()` et `repaint()` pour les mises à jour.

2.3 Diagramme de séquence

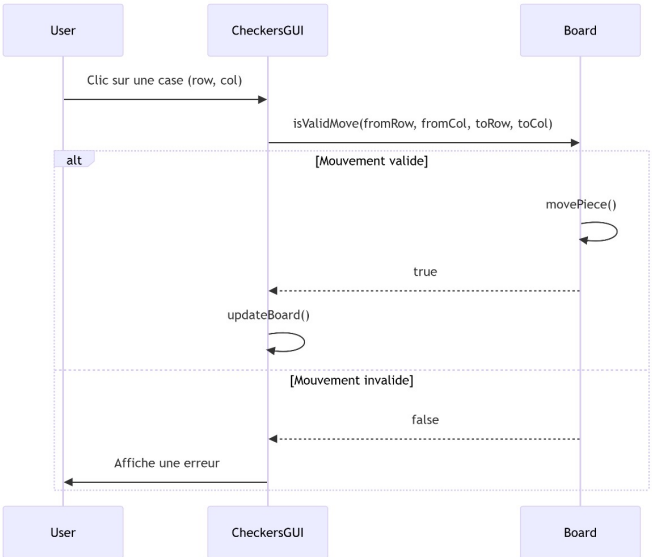

FIGURE 2.1 – Diagramme de séquence

2.4 Diagramme de Classes UML

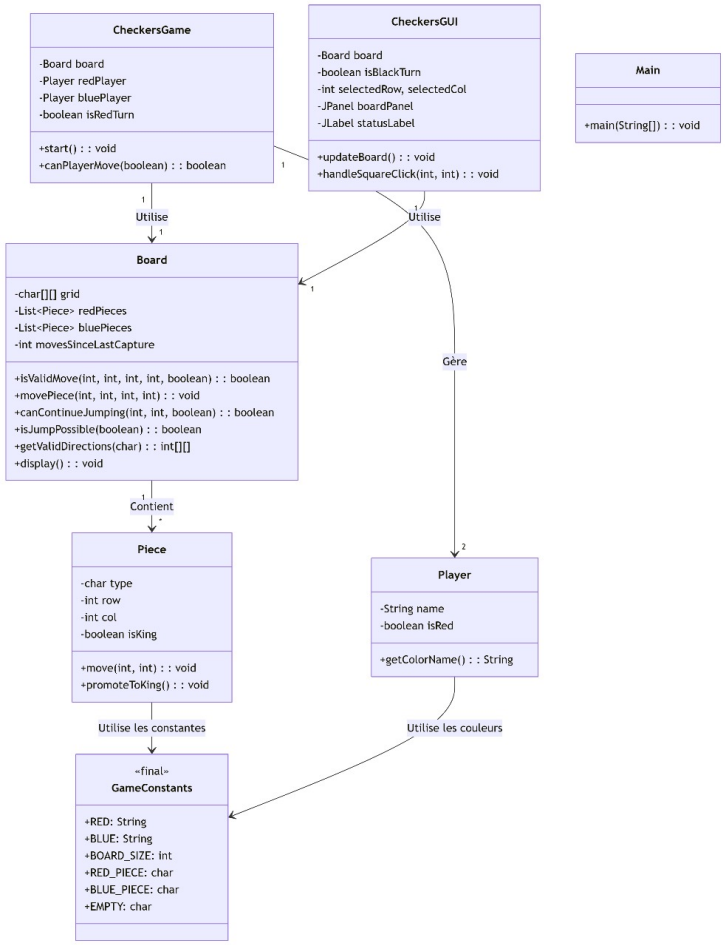

FIGURE 2.2 – Diagramme de Classes

Chapitre 3

Illustrations, Répartition du Travail et Bilan du Projet

3.1 Captures d'écran du jeu

Voici quelques captures d'écran illustrant l'exécution du jeu de dames :

FIGURE 3.1 – Page d'accueil du jeu. Cette interface permet au joueur de démarrer une nouvelle partie ou de consulter les options.

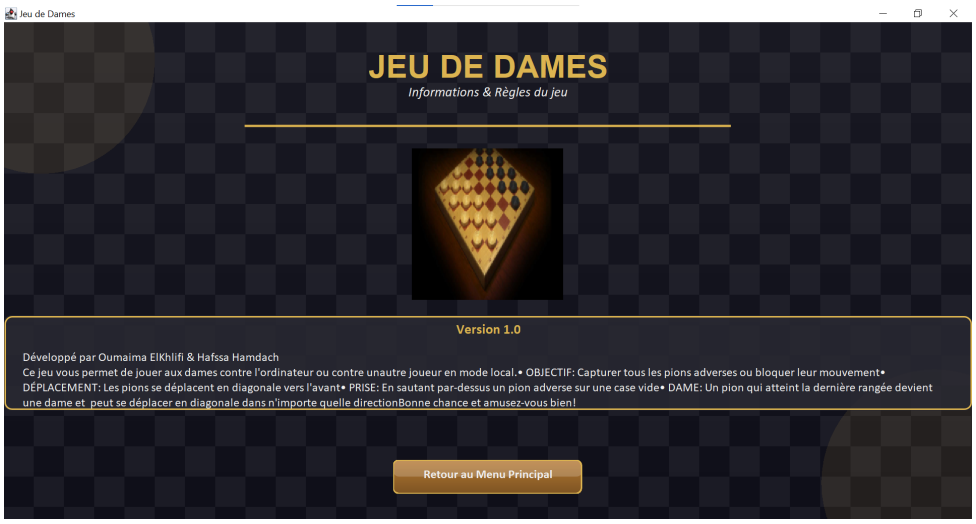

FIGURE 3.2 – Page d'information du jeu. Elle présente les règles, les objectifs et les instructions pour comprendre le fonctionnement du jeu.

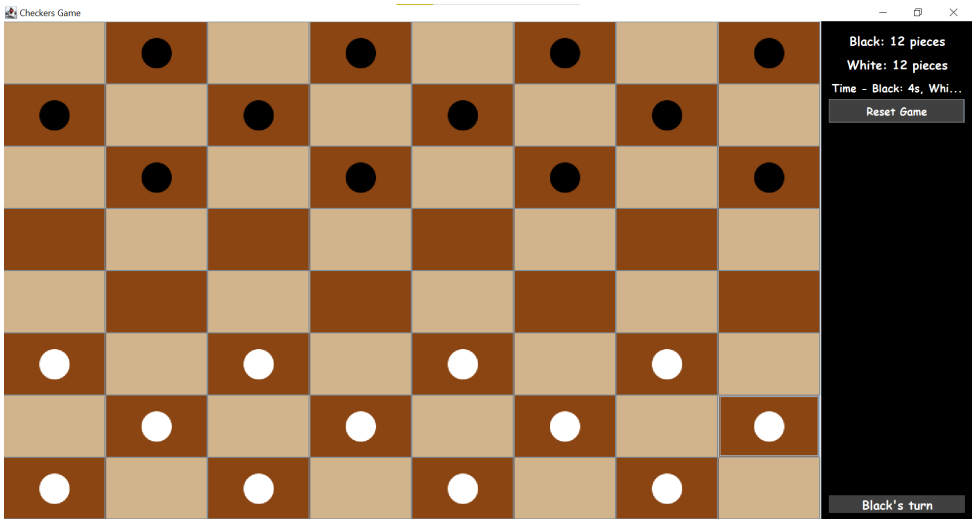

FIGURE 3.3 – Plateau de jeu en cours. Le joueur visualise la disposition des pions et peut effectuer des déplacements.

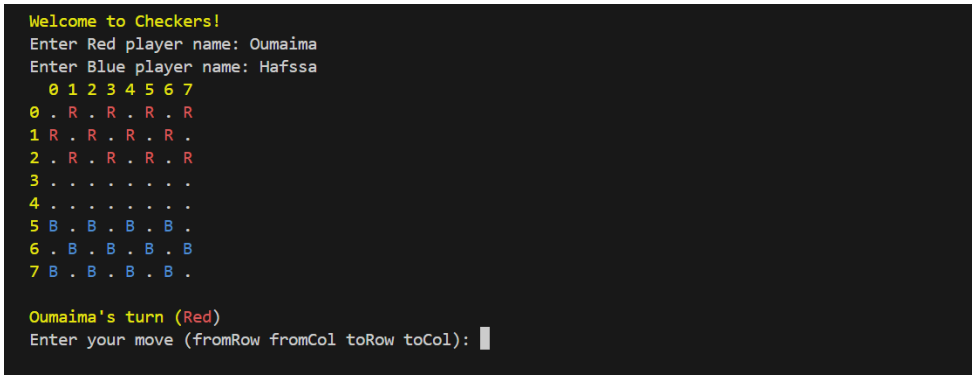

FIGURE 3.4 – Interface console du jeu. Cette version textuelle du jeu permettait de tester la logique des déplacements et captures avant l'intégration graphique.

3.2 Difficultés Rencontrées et Améliorations Possibles

Difficultés rencontrées

- Gestion des règles complexes du jeu
- Coordination entre la logique du jeu et l'interface graphique
- Débogage difficile
- Répartition du travail entre les développeurs

Améliorations possibles

- Ajout d'un système de sauvegarde
- Amélioration de l'interface Swing (animations, couleurs, menu)
- Ajout d'un mode joueur contre IA
- Mise en place de tests automatisés

3.3 Répartition des tâches

Membre	**Tâches effectuées**
Hafssa Hamdach	Développement de l'interface en mode console.
Oumaima El Khlifi	Développement de l'interface graphique avec Swing.
Les deux	Collaboration sur la réalisation du PPT, du rapport du projet, ainsi que la conception du diagramme de classe et diagramme de séquence.