
SPÉCIFICATIONS

INFO SI - MEDEV

Réalisé par :
Fernando ROJAS
Augusto ARROJO

Année scolaire: 2024/2025

Table des matières

Table des matières	2
1. Objectif du projet	3
2. Exigences du Projet	3
2.1. Exigences Fonctionnelles	3
2.2. Exigences Non Fonctionnelles	3
2.3. Contraintes	3
3. Conception du système	4
3.1. Diagramme de cas d'utilisation	4
3.2. Diagramme de classes	4
3.3. Architecture	4
3.4. Flux de travail	4
4. Technologies et outils	4
5. Planification	5
6. Annexes	5

1. Objectif du projet

1.1. Introduction

Le projet consiste à développer un jeu d'Othello (Reversi) en Java avec une interface texte permettant à deux joueurs de s'affronter. Le programme doit être structuré, clair et respecter les standards de codage. Les commentaires doivent être en français pour expliquer le fonctionnement du code.

1.2. Fonctionnalités requises

1.2.1. Initialisation du jeu

Un plateau de jeu de 8x8 cases doit être créé.

Les quatre pions initiaux doivent être placés au centre du plateau : deux noirs (B) en e4 et d5, et deux blancs (W) en d4 et e5.

1.2.2. Affichage du plateau

Le plateau doit être affiché à chaque tour, avec les colonnes identifiées par des lettres (a à h) et les lignes par des chiffres (1 à 8).

Les cases vides doivent être représentées par un tiret (-).

1.2.3. Placement des pions

Chaque joueur peut placer un pion sur une case valide selon les règles du jeu d'Othello.

Un placement est valide si au moins un pion adverse est capturé.

La capture implique que les pions adverses pris en sandwich entre deux pions du joueur actuel changent de couleur.

1.2.4. Gestion des tours

Noir commence toujours la partie.

Les joueurs jouent à tour de rôle.

Si un joueur ne peut pas effectuer de mouvement valide, il doit passer son tour.

1.2.5. Fin de la partie

La partie se termine si aucun des joueurs ne peut jouer ou si le plateau est rempli.

Le joueur ayant le plus grand nombre de pions de sa couleur gagne la partie.

1.2.6. Validation des entrées utilisateur

Le programme doit vérifier les entrées des joueurs pour s'assurer qu'elles sont valides (par exemple, "d3").

Si une entrée est invalide, le programme doit demander au joueur de réessayer.

2. Exigences du Projet

2.1. Exigences Fonctionnelles

- Permettre à deux joueurs de jouer à tour de rôle sur un othellier de 8x8 cases.
- Valider les mouvements des joueurs selon les règles d'Othello.
- Capturer les pions adverses lorsqu'un mouvement est valide.
- Gérer automatiquement le changement de tour entre les joueurs.
- Afficher le plateau après chaque tour.
- Détecter la fin de la partie et annoncer le gagnant.

2.2. Exigences Non Fonctionnelles

- Temps de réponse des actions utilisateur inférieur à 2 secondes.
- Supporte un maximum de 2 joueurs simultanément.
- Compatible avec Java 17+.

2.3. Contraintes

- Le projet doit être développé en Java 17+.

3. Conception du système

3.1. Diagramme de cas d'utilisation

3.2. Diagramme de classes

Le diagramme montre deux acteurs :
Joueur 1 et Joueur 2.

Actions principales : Jouer un tour, valider un mouvement, terminer une partie.

3.3. Architecture

- Présentation : Interface texte par console.
- Logique métier : Gestion des emprunts et des utilisateurs.
- Données : Stockage local dans le dispositif.

3.4. Flux de travail

Déroulement d'un tour :

- Afficher le plateau.
- Demander une action au joueur actuel.
- Valider l'action.
- Capturer les pions si le mouvement est valide.
- Passer au joueur suivant.

4. Technologies et outils

- Langage : Java 18
- IDE : Netbeans
- Tests : JUnit
- Gestion de version : Git (GitHub)
- Automatisation : Ant Apache
- Qualité du code : SonarQube

5. Planification

- Semaine 1 : Analyse des exigences.

- Semaines 2-3 : Conception du système.
- Semaines 4-6 : Développement des fonctionnalités principales.
- Semaine 7 : Tests et intégration.
- Semaine 8 : Documentation et livraison.

6. Annexes