

Développement du Jeu Gomoku en Java

Boudjelel Mahdi
Date : 9 mai 2025

Lien du dépôt GitHub

Le projet est disponible sur GitHub à l'adresse suivante :
<https://github.com/Divert12/JeuGomuko.git>

Introduction

Ce projet a pour but de développer une version console du jeu Gomoku en Java. Le jeu se joue à deux, avec une grille personnalisable et un nombre limité de jetons. L'objectif est d'aligner un nombre défini de jetons horizontalement, verticalement ou en diagonale.

Architecture du code

Le projet est organisé en plusieurs classes pour favoriser une architecture propre et modulaire.

- **Case** : Représente chaque case de la grille, contient un joueur, et les références à ses voisins dans 8 directions.
- **Direction** : Enumération représentant les 8 directions autour d'une case (N, NE, E, SE, S, SO, O, NO).
- **Joueur** : Contient les informations d'un joueur (nom, couleur, jetons restants).
- **Grille** : Matrice de cases. Elle relie les voisins de chaque case et permet de vérifier la présence de jetons.
- **Gomoku** : Contient la logique du jeu (placement des jetons, changement de joueur, détection de victoire, sauvegarde/chargement).
- **Affichage** : Gère l'affichage console de la grille avec couleurs ANSI.
- **Main** : Point d'entrée principal, gère les interactions utilisateur via la console.

La méthode `estDansAlignement` dans la classe `Case` vérifie si un joueur a gagné, en parcourant les directions et leur opposé pour calculer un alignement.

Compilation - Exécution

Compilation

Pour compiler le projet, utilisez la commande suivante (en supposant que les sources sont dans le dossier `src`) :

```
javac -d compiled/ src/**/*.java src/Main.java
```

Exécution

Exécution avec les paramètres par défaut (grille 15x15, 60 jetons, alignement de 5) :

```
java -cp compiled/ Main
```

Exécution avec paramètres personnalisés :

```
java -cp compiled/ Main [taille] [jetons] [alignement]
```

Robustesse

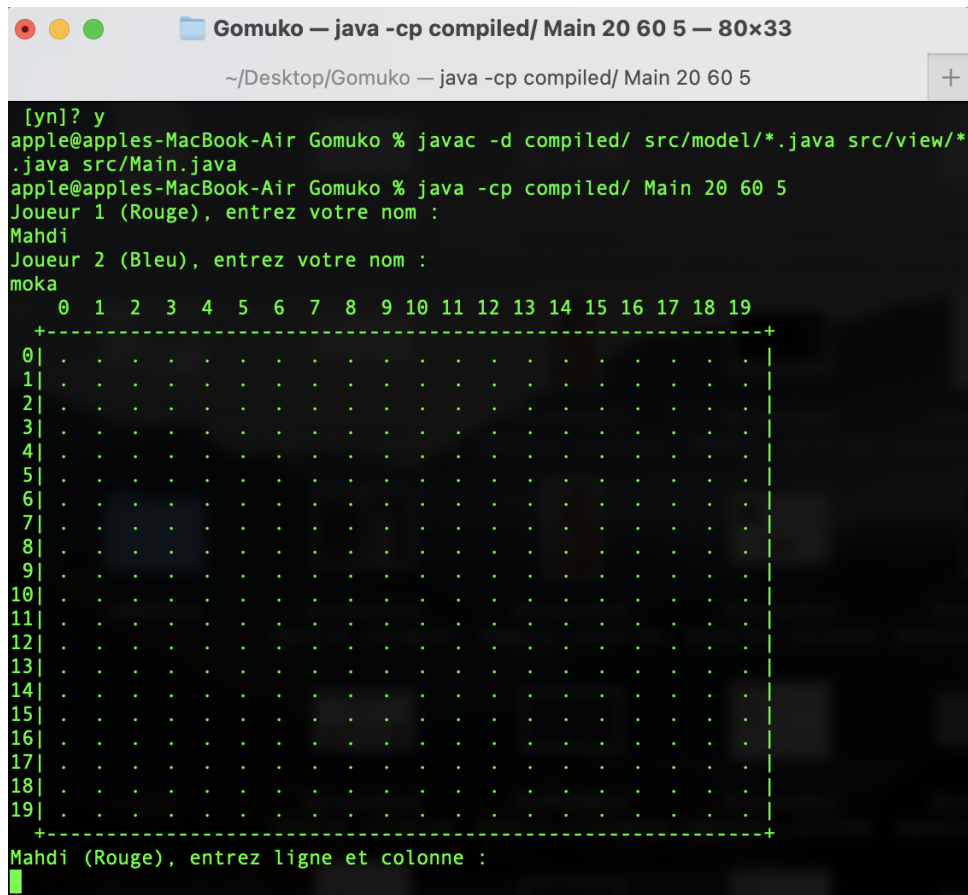
Des exceptions personnalisées comme `CoupInvalideException` sont utilisées pour séparer la logique métier des erreurs de jeu.

Sauvegarde/Chargement

Le jeu peut être sauvegardé dans un fichier texte via un format simple :

- Paramètres de la partie (taille, condition de victoire).
- État des joueurs.
- Grille sous forme de caractères (R, B, .).

Illustration du Jeu



```
Gomuko — java -cp compiled/ Main 20 60 5 — 80x33
~/Desktop/Gomuko — java -cp compiled/ Main 20 60 5

[yn]? y
apple@apples-MacBook-Air Gomuko % javac -d compiled/ src/model/*.java src/view/*.java src/Main.java
apple@apples-MacBook-Air Gomuko % java -cp compiled/ Main 20 60 5
Joueur 1 (Rouge), entrez votre nom :
Mahdi
Joueur 2 (Bleu), entrez votre nom :
moka
  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
+-----+
0| . . . . . . . . . . . . . . . . . . . . . .
1| . . . . . . . . . . . . . . . . . . . . . .
2| . . . . . . . . . . . . . . . . . . . . . .
3| . . . . . . . . . . . . . . . . . . . . . .
4| . . . . . . . . . . . . . . . . . . . . . .
5| . . . . . . . . . . . . . . . . . . . . . .
6| . . . . . . . . . . . . . . . . . . . . . .
7| . . . . . . . . . . . . . . . . . . . . . .
8| . . . . . . . . . . . . . . . . . . . . . .
9| . . . . . . . . . . . . . . . . . . . . . .
10| . . . . . . . . . . . . . . . . . . . . . .
11| . . . . . . . . . . . . . . . . . . . . . .
12| . . . . . . . . . . . . . . . . . . . . . .
13| . . . . . . . . . . . . . . . . . . . . . .
14| . . . . . . . . . . . . . . . . . . . . . .
15| . . . . . . . . . . . . . . . . . . . . . .
16| . . . . . . . . . . . . . . . . . . . . . .
17| . . . . . . . . . . . . . . . . . . . . . .
18| . . . . . . . . . . . . . . . . . . . . . .
19| . . . . . . . . . . . . . . . . . . . . . .
+-----+
Mahdi (Rouge), entrez ligne et colonne :
█
```

Cette capture d'écran montre un exemple d'exécution dans la console, avec les jetons rouges et bleus colorés selon les joueurs.

Conclusion

Ce projet de Gomoku a été l'occasion de mettre en œuvre une architecture claire et orientée objet, avec une séparation des responsabilités efficace. Le système est extensible, lisible et intègre les principes de bonne conception logicielle.