

SquareGame — Mini■cartographie du projet

Résumé fonctionnel et architectural (v1.1.7)

Vision globale

App.vue orchestre l'interface utilisateur. Les composables portent la logique métier. Les fichiers de types et constantes constituent la fondation TypeScript.

Niveaux de responsabilité

- App.vue — orchestration UI, événements, affichage
- Composables — logique métier isolée et réutilisable
- Types / Constantes — fondation pure TypeScript

Détail des fichiers principaux

App.vue

Chef d'orchestre UI. Assemble les composables, gère l'état d'interface, déclenche les actions utilisateur et les animations. Ne contient aucune logique métier.

useBoard.ts

Source de vérité du jeu. Contient l'état du plateau, l'historique des coups et les actions fondamentales (play, undo, reset). Toute mutation du jeu passe par ce fichier.

useMoves.ts

Règles de déplacement. Calcule les coups valides et détecte les impasses. Fonctions dérivées, sans effet de bord.

useTip.ts

Aide locale basée sur Warnsdorff. Calcule un coup conseillé sans modifier le plateau. Instantané et sûr.

useSolver.ts

Moteur de recherche complet. Implémente DFS + heuristique de Warnsdorff avec cutoff. Totalement indépendant de Vue et de l'UI.

useSolveRunner.ts

Pont entre le solveur et l'interface. Anime l'exécution pas à pas, gère les délais et protège contre les actions concurrentes.

moveOffsets.ts

Définition géométrique des déplacements. Constantes pures et réutilisables.

BoardState.ts

Définition des types fondamentaux (Pos, BoardState). Aucune dépendance Vue.

Règles d'or

- Une seule source de vérité pour l'état du jeu
- Logique métier hors des composants Vue
- UI réactive, jamais décisionnelle
- Animations gérées par le CSS quand c'est possible

Conclusion — SquareGame constitue une base saine, modulaire et pédagogique, réutilisable pour d'autres applications Vue + TypeScript.