Plan de tests

Par : **Cédric MAS**
Thomas LEPAGE
David KIZAYILAWOKO

20 octobre 2025

Sommaire

1 Introduction

1.1 Présentation du projet

Le projet consiste en une implémentation du jeu Connect Four (Puissance 4), incluant un moteur de jeu en TypeScript et une interface React pour l'interaction utilisateur. Le jeu permet à deux joueurs (Rouge et Jaune) de placer des pièces dans une grille de 6x7, avec détection de victoires horizontales, verticales et diagonales.

1.2 Objectifs du plan de test

Ce plan de test vise à valider les fonctionnalités principales du moteur de jeu et de l'interface utilisateur, en s'appuyant sur les tests unitaires implémentés dans les fichiers `engine.test.ts` et `App.test.tsx`. Il évalue la conformité des scénarios, identifie les anomalies potentielles et assure la robustesse du jeu.

2 Contexte du projet

2.1 Description du contexte

Le jeu est développé en TypeScript avec Vitest pour les tests unitaires et React pour l'UI. Les tests couvrent le placement de pièces, la détection de victoires et les interactions UI de base.

2.2 Analyse des besoins

Les besoins incluent : placement valide de pièces, détection précise des victoires, rendu correct de la grille, gestion des tours et réinitialisation du jeu.

3 Objectifs du projet

3.1 Objectifs généraux

Assurer que le jeu respecte les règles de Connect Four et fournit une expérience utilisateur fluide.

3.2 Objectifs spécifiques

- Valider le placement de pièces dans les colonnes.
- Détecter les victoires dans toutes les directions.
- Tester le rendu et les interactions de l'interface React.

4 Fonctionnalités requises

4.1 Liste exhaustive des fonctionnalités

- Placement de pièces dans une colonne (bas vers haut).
- Détection de victoires (horizontale, verticale, diagonale, anti-diagonale).
- Rendu de la grille et indication du joueur courant.
- Aperçu de pièce au survol.
- Détection de match nul et victoires.
- Réinitialisation du jeu.

4.2 Priorisation des fonctionnalités

Priorité haute : Placement et détection de victoires.
Priorité moyenne : Interactions UI (survol, clic).
Priorité basse : Alertes de fin de jeu.

4.3 Interactions entre les fonctionnalités

Le placement déclenche la vérification de victoire et le changement de tour. L'UI réagit aux états du moteur.

5 Contraintes et limitations

5.1 Contraintes de temps

Tests exécutés le 30 septembre 2025.

5.2 Contraintes techniques

Tests limités aux environnements Vitest et React Testing Library. Pas de tests e2e pour l'instant.

6 Tests et validation

6.1 Stratégie de test

Tests unitaires pour le moteur (Engine) et composants React (App). Utilisation de mocks pour isoler les dépendances. Couverture : placement, victoires, rendu UI, réinitialisation.

6.2 Critères de réussite des tests

- Tous les tests passent sans erreurs.
- Conformité : Résultat observé matches le résultat attendu.
- En attente : Tests commentés ou incomplets.
- Non-conforme : Échecs ou anomalies détectées.

6.3 Procédure de validation du projet

Exécuter `vitest` sur les fichiers de test. Analyser les résultats et corriger les anomalies.

7 Glossaire

7.1 Définition des termes techniques

- Engine : Classe gérant la logique du jeu (placement, victoires).
- Vitest : Framework de test utilisé.
- Conforme : Test passe avec succès.
- En attente : Test non implémenté ou commenté.

Récapitulatif des Tests

Scénario	Résultat attendu	Date du test	État	Anomalie	Correction	Nombre d'essai
Placement d'une pièce dans la cellule vide la plus basse d'une colonne	Pièce placée à {row: 5, col: 0, color: Red}	30/09/2025	Conforme	Aucune	N/A	1
Placement au-dessus d'une pièce existante dans la même colonne	Pièce placée à {row: 4, col: 0, color: Yellow}	30/09/2025	Conforme	Aucune	N/A	1
Retour null pour une colonne pleine	Null retourné après 6 placements	30/09/2025	Conforme	Aucune	N/A	1
Retour null pour une colonne invalide	Null pour colonne 7	30/09/2025	Conforme	Aucune	N/A	1
Faux pour une cellule vide	False pour cellule null	30/09/2025	Conforme	Aucune	N/A	1
Détection de victoire horizontale	True pour 4 Red horizontaux	30/09/2025	Conforme	Aucune	N/A	1
Détection de victoire verticale	True pour 4 Yellow verticaux	30/09/2025	Conforme	Aucune	N/A	1
Détection de victoire diagonale (top-left to bottom-right)	True pour 4 Red diagonaux	30/09/2025	Conforme	Aucune	N/A	1
Détection de victoire anti-diagonale (top-right to bottom-left)	True pour 4 Yellow anti-diagonaux	30/09/2025	Conforme	Aucune	N/A	1
Faux pour moins de 4 en ligne	False pour 3 Red horizontaux	30/09/2025	Conforme	Aucune	N/A	1
Rendu du plateau et joueur courant	Titre et 42 cellules rendues	30/09/2025	En attente	Tests supplémentaires en attente	Ajouter tests hover/clic	0
Réinitialisation du jeu sur clic bouton	Jeu reset, alertes supprimées	30/09/2025	En attente	Tests supplémentaires en attente	Vérifier état après reset	0

Nombre de tests : 12 (10 pour Engine, 2 pour App actifs ; d'autres commentés en attente).
Conforme : 10 (83%)
En attente : 2 (17%)
Non-conforme : 0 (0%)