

Cahier des charges

pour le Projet de Jeu Connect Four

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

20 octobre 2025

Sommaire

1	Introduction	3
1.1	Présentation du projet	3
1.2	Objectifs du cahier des charges	3
2	Contexte du projet	3
2.1	Description du contexte	3
2.2	Analyse des besoins	3
3	Objectifs du projet	3
3.1	Objectifs généraux	3
3.2	Objectifs spécifiques	3
4	Fonctionnalités requises	3
4.1	Liste exhaustive des fonctionnalités	3
4.2	Priorisation des fonctionnalités	4
4.3	Interactions entre les fonctionnalités	4
5	Contraintes et limitations	4
5.1	Contraintes de temps	4
5.2	Contraintes techniques	4
6	Tests et validation	4
6.1	Stratégie de test	4
6.2	Critères de réussite des tests	4
6.3	Procédure de validation du projet	4
7	Glossaire	4
7.1	Définition des termes techniques utilisés dans le cahier des charges	4

1 Introduction

1.1 Présentation du projet

Le projet consiste en le développement d'un jeu Connect Four (Puissance 4) en ligne, implémenté en TypeScript avec React pour l'interface utilisateur. Le jeu oppose deux joueurs qui placent alternativement des jetons rouges ou jaunes dans une grille de 6 lignes et 7 colonnes. L'objectif est d'aligner quatre jetons de la même couleur horizontalement, verticalement ou en diagonale.

Le moteur de jeu gère la logique (placement des pièces, détection des victoires), tandis que l'interface permet une interaction intuitive via clics et survols de souris.

1.2 Objectifs du cahier des charges

Ce document vise à définir précisément les exigences fonctionnelles et non fonctionnelles du projet, à identifier les contraintes, et à établir une stratégie de test et de validation. Il sert de référence pour le développement, les tests et la validation finale.

2 Contexte du projet

2.1 Description du contexte

Ce projet s'inscrit dans le cadre d'un exercice de développement logiciel, simulant un environnement de production pour un jeu web simple. Il utilise des technologies modernes comme React pour l'UI et Vitest pour les tests unitaires. Le jeu est conçu pour être joué localement par deux joueurs alternant sur le même appareil.

2.2 Analyse des besoins

Les besoins identifiés incluent :

- Une interface utilisateur réactive et intuitive.
- Une logique de jeu robuste pour gérer les placements et les victoires.
- Des tests unitaires couvrant les fonctionnalités clés.
- Une gestion des états de jeu (tour du joueur, victoire, match nul).

3 Objectifs du projet

3.1 Objectifs généraux

Développer un jeu Connect Four fonctionnel, accessible via un navigateur web, respectant les règles classiques du jeu.

3.2 Objectifs spécifiques

- Implémenter un moteur de jeu capable de placer des pièces et de détecter les alignements de quatre jetons.
- Créer une interface avec affichage de la grille, prévisualisation des placements, et messages de fin de partie.
- Assurer la compatibilité avec les navigateurs modernes.
- Couvrir au moins 80% du code par des tests unitaires.

4 Fonctionnalités requises

4.1 Liste exhaustive des fonctionnalités

- Initialisation d'une nouvelle partie avec une grille vide et un joueur aléatoire commençant.
- Placement d'un jeton dans une colonne choisie, en respectant la gravité (tombe au bas de la colonne).
- Alternance automatique des tours entre joueurs Rouge et Jaune.
- Détection de victoire (horizontal, vertical, diagonal, anti-diagonal).
- Détection de match nul lorsque la grille est pleine sans vainqueur.
- Prévisualisation du jeton au survol d'une colonne.
- Bouton de réinitialisation pour recommencer une partie.

- Affichage du joueur courant et des messages de victoire ou nul.

4.2 Priorisation des fonctionnalités

- Priorité haute : Logique de placement, détection de victoire, alternance des tours.
- Priorité moyenne : Prévisualisation au survol, messages de fin de partie.
- Priorité basse : Animations avancées (non implémentées dans la version actuelle).

4.3 Interactions entre les fonctionnalités

- Le placement d'un jeton déclenche la vérification de victoire et, si nécessaire, l'alternance des tours.
- La prévisualisation dépend de l'état du joueur courant et s'active uniquement si la partie n'est pas terminée.
- La réinitialisation rétablit l'état initial, effaçant la grille et choisissant un nouveau joueur de départ.

5 Contraintes et limitations

5.1 Contraintes de temps

Le projet doit être complété dans un délai de développement simulé de 2 semaines, avec des milestones pour l'implémentation du moteur et de l'UI.

5.2 Contraintes techniques

- Technologies imposées : TypeScript, React, Vitest pour les tests.
- Pas d'accès à des bibliothèques externes pour la logique de jeu (tout implémenté from scratch).
- Compatibilité avec les navigateurs web standards (Chrome, Firefox).
- Limitation à un mode deux joueurs local (pas de multiplayer en ligne).

6 Tests et validation

6.1 Stratégie de test

- Tests unitaires pour le moteur de jeu (placement, détection de victoire) via Vitest.
- Tests d'intégration pour l'interface React (clics, survols, affichage).
- Tests manuels pour les scénarios de victoire, nul, et colonnes pleines.

6.2 Critères de réussite des tests

- Tous les tests unitaires passent sans erreur.
- Couverture de code supérieure à 80%.
- Absence d'anomalies critiques (crashes, placements invalides).
- Validation manuelle : Le jeu respecte les règles de Connect Four dans tous les cas testés.

6.3 Procédure de validation du projet

1. Exécution des tests automatisés.
2. Tests manuels par l'équipe de développement.
3. Revue de code pour vérifier le respect des bonnes pratiques.
4. Validation finale : Démonstration d'une partie complète sans bugs.

7 Glossaire

7.1 Définition des termes techniques utilisés dans le cahier des charges

- **Grille** : Matrice 6x7 représentant le plateau de jeu.
- **Jeton** : Pièce colorée (Rouge ou Jaune) placée dans la grille.
- **Victoire** : Alignement de quatre jetons identiques.
- **Match nul** : Grille pleine sans victoire.
- **Prévisualisation** : Affichage temporaire d'un jeton au survol.
- **Moteur de jeu** : Classe Engine gérant la logique interne.