

# Cahier des Charges : Jeu de Dames Fonctionnel sur Unity

## 1. Présentation du Projet

Nom du Projet : Jeu de Dames en Local

Description : Développement d'un jeu de dames fonctionnel en mode joueur contre joueur sur le même ordinateur, utilisant Unity comme moteur de jeu.

Objectif : Offrir une expérience fluide et respectant les règles officielles du jeu de dames. Le projet inclut le développement, les tests unitaires, la documentation et une gestion efficace sur GitHub.

## 2. Contexte

Le projet s'inscrit dans le cadre d'un travail collaboratif en groupe (3 à 4 étudiants) visant à renforcer les compétences en développement logiciel, gestion de projet, et pratiques de collaboration sur GitHub.

## 3. Fonctionnalités du Jeu

- Interface utilisateur :
  - Plateau de jeu 8x8 avec cases alternées (noires et blanches).
  - Pièces distinctes pour chaque joueur (blanches et noires).
  - Système d'interaction intuitif pour déplacer les pièces (clic souris ou glisser-déposer).
- Mécanique de Jeu :
  - Respect des règles officielles du jeu de dames (déplacement en diagonale, captures obligatoires, promotion en dame, etc.).
  - Alternance automatique des tours entre les deux joueurs.
  - Vérification des coups valides et indication des mouvements possibles.
- Fonctionnalités avancées :
  - Détection de la fin de partie (victoire, match nul, blocage).
  - Option de réinitialisation de la partie.
  - Possibilité d'annuler le dernier coup (optionnelle).
- Design :
  - Plateau visuellement attrayant avec animations fluides pour les déplacements.
  - Indications visuelles des pièces sélectionnées et des mouvements possibles.

## 4. Spécifications Techniques

- Moteur de jeu : Unity.

- Langage de programmation : C#.
- Framework de tests unitaires : NUnit intégré à Unity.
- Gestion des versions : GitHub.
- Normes de développement : Respect des conventions C#.

## 5. Plan de Réalisation

1. Analyse et conception :
  - Comprendre les règles officielles du jeu de dames.
  - Concevoir une architecture du jeu (modèle MVC ou similaire).
  - Planifier les tâches en sous-tâches sur GitHub (issues et branches).
2. Développement :
  - Implémenter les fonctionnalités de base (plateau, pièces, mouvements valides).
  - Ajouter les règles avancées (captures, promotion en dame, fin de partie).
3. Tests et validation :
  - Rédiger et exécuter des tests unitaires pour valider les règles de jeu.
  - Effectuer des tests fonctionnels sur différentes machines.
4. Documentation :
  - Rédiger une documentation technique détaillée.
  - Écrire un plan de test décrivant les scénarios testés.
5. Collaboration et revue :
  - Effectuer des pull-requests régulières pour chaque fonctionnalité.
  - Réaliser des revues de code sur GitHub.

## 6. Critères d'Acceptation

- Respect des règles officielles du jeu de dames.
- Interface utilisateur intuitive et responsive.
- Code propre, lisible et bien commenté.
- Couverture complète des fonctionnalités via des tests unitaires.
- Documentation claire, complète et conforme au projet.

## 7. Plan de Test

- Scénarios de test :
  - Vérification du déplacement des pièces sur le plateau.
  - Validation des captures obligatoires.
  - Promotion correcte en dame.
  - Détection de la fin de partie (victoire, nul, blocage).
- Critères d'acceptation :
  - Toutes les règles sont respectées.

- Aucun crash ou comportement imprévu pendant une partie.
- Interface fluide et réactive.

## 8. Outils et Technologies

- Unity pour le développement.
- Visual Studio pour l'écriture du code.
- NUnit pour les tests unitaires.
- Git et GitHub pour la gestion de versions.
- Canva/Illustrator pour le design des assets visuels (si nécessaire).

## 9. Gestion de Projet

- Organisation via Trello ou GitHub Projects pour suivre l'avancement.
- Répartition des rôles :
  - Développeur principal : Responsable de l'implémentation des règles du jeu.
  - Designer : Chargé de l'esthétique et de l'interface.
  - Testeur : Responsable de l'écriture des tests unitaires et fonctionnels.
  - Documentaliste : Prépare la documentation technique et le cahier des charges.

## 10. Livrables

- Code source complet sur GitHub.
- Documentation technique détaillée.
- Cahier des charges.
- Plan de test.
- Rapport final incluant les retours sur le développement.

## 11. Contraintes

- Respect des normes de codage et des conventions C#.
- Projet livré dans les délais impartis.
- Tests couvrant au minimum 80% des fonctionnalités.