

# Documentation Technique

Jeu de dames



Jayson Poncey Valdemar, Alix Silvani, Léa Parent, Nathan Bugeaud

# Sommaire

1. Introduction .....	3
2. Technical Design .....	4
3. Project Architecture.....	7
6. Project Structure .....	7
4. Wireframe / Mock Up .....	8
5. User structure .....	9
7. API documentation .....	10
8. Glossaire .....	10

# Introduction

## Objectif du projet

Le jeu de dames est un jeu de stratégie classique apprécié dans le monde entier. Ce projet vise à créer une version numérique de ce jeu, permettant aux utilisateurs de jouer contre un autres joueurs en local. Ce jeu se veut accessible, intuitive et conforme aux règles officielles du jeu de dames.

## Portée de la documentation

Cette documentation technique est conçue pour les développeurs et les parties prenantes du projet. Elle offre une vue d'ensemble complète du système, depuis la conception du moteur de jeu jusqu'aux interfaces utilisateur.

## Contexte général

Le projet repose sur une architecture moderne utilisant C# en utilisant le modèle MVC (Model View Controller) pour la réalisation du jeu

---

# Technical Design

## Règles de nommage

Pour le projet les règles de nommage utilisées sont le CamelCase et le PascalCase

Exemple utilisation du PascalCase:

PascalCase

- Définition : Chaque mot commence par une majuscule, y compris le premier.
- Utilisé pour :
  - Noms de classes : GameController
  - Méthodes : GetPrevButtonColor
  - Propriétés : public Board Board { get; set; }
  - Noms d'espaces de noms :
    - MyNamespace.SubNamespace
  - Énumérations : Color.Red

Exemple utilisation du camelCase:

- Définition : Le premier mot commence par une minuscule, mais les mots suivants commencent par une majuscule.
- Utilisé pour :
  - Variables locales : bool player1 = false;
  - Paramètres de méthodes : Button prevButton de
    - (GetPrevButtonColor)
  - Champs privés (souvent avec un préfixe `_`) :  
\_lblVictoryMessage

## Règles de commit

Pour les commits du projet on mettait un titre simple et claire et une description plus détaillé de ce qui avait été fait exemple;

Titre: Ajout Restart

Description: Ajout d'un bouton restart pour relancer le jeu

## Règles de Design Pattern

Pour le jeu on à utiliser le MVC (Model View Controller), on l'a utiliser dans le but de clarifier le projet et de le rendre plus lisible le but derrière est de favoriser et accélérer la compréhension des différents développeur afin qu'il puisse comprendre ce qui a été fait. Exemple de l'utilisation du MVC dans notre projet:

- Model : Gère la logique métier (règles du jeu, position des pions, calcul des mouvements).
- View : Affiche l'état du jeu à l'utilisateur (plateau, pions, messages).
- Controller : Relie les deux, en interprétant les actions de l'utilisateur (clics sur le plateau, choix des mouvements).

De même pour la testabilité du jeu qui en est accrue car on peut tester les différentes parties sans toucher aux autres de même pour la maintenance du jeu ainsi que sa modification par exemple passer de la 2D à la 3D en modifiant que la View. La gestion des interactions utilisateur aussi est améliorée car elle est simplifiée et permet l'ajout de nouvelles fonctionnalités facilement.

## Commits

Pour les commits du projet on mettait un titre simple et claire et une description plus détaillé de ce qui avait été fait exemple;

Titre: Ajout Restart

Description: Ajout d'un bouton restart pour relancer le jeu

## Test

Pour les tests un projet à été ajouté CheckGameTestUnitaire permettant de faire des tests sur chaque fichier pour tester l'intégralité des fonction sous la forme:

Nom du test et son utilité / le résultat attendu / date du test / si il est conforme ou pas

si celui-ci n'est pas conforme il faut expliquer l'anomalie et la solution pour résoudre cette anomalie

# Project Architecture

## Choix des technologies

Pour le projet certaines technologies ont été utilisé, pour le langage c'est le C# qui à été utilisé pour différente raison, la première est que le C# est adapté pour la conception de jeu vidéo tel que notre jeu de dame la suivante est l'utilisation graphique de Windows Form qui nous a permit d'obtenir un visuel graphique pour jouer au jeu de dame

## Project Structure

### CheckGame

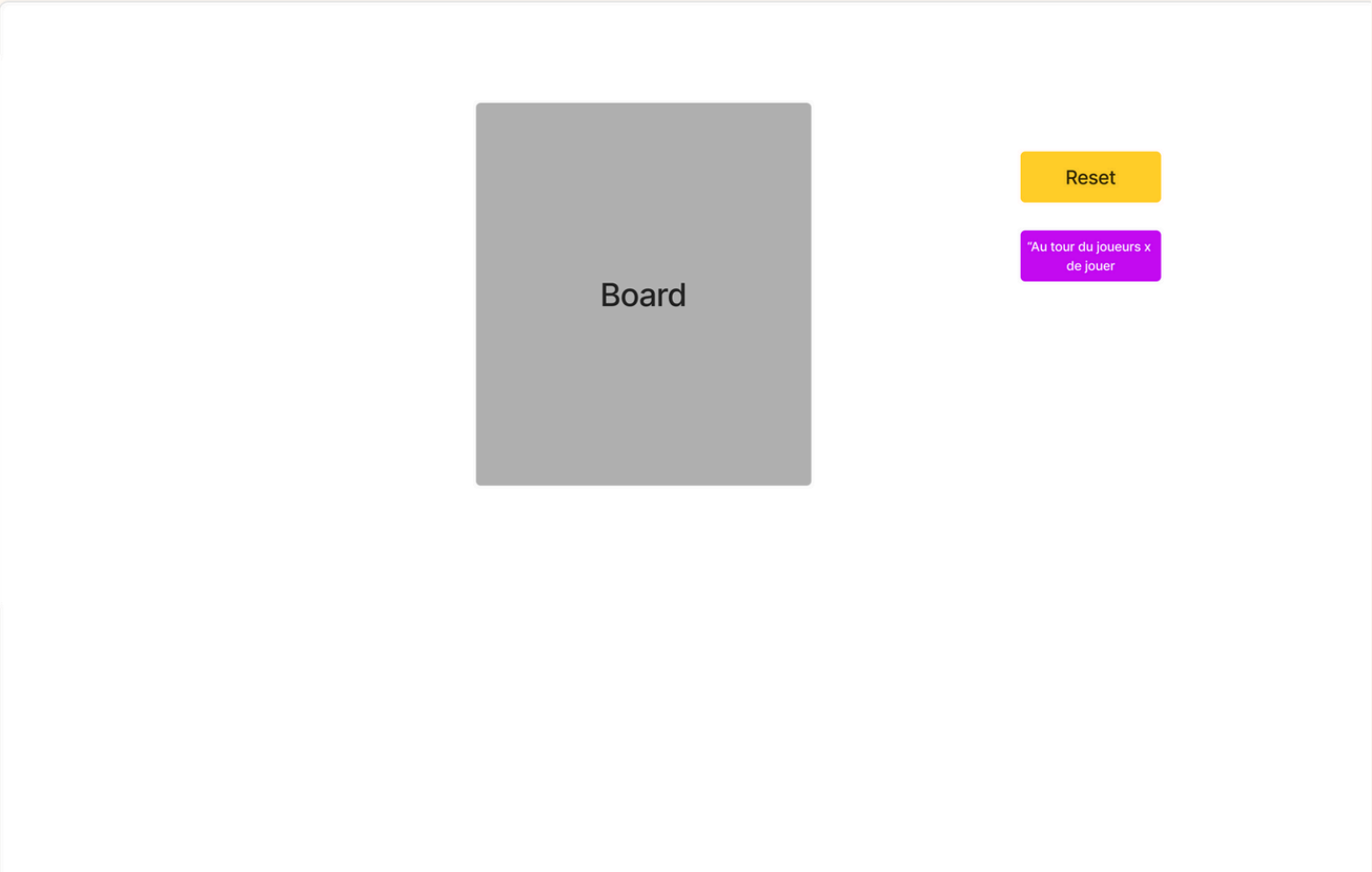
- └─Dépendance //Bibliothèque externe ou interne
- └─Analyseurs //Permet de vérifier les bonnes pratiques/erreurs
- └─Frameworks //Répertoire où se trouve les Frameworks
- └─Controllers //Répertoire contenant la logique du jeu
- └─GameController.cs //Fichier contenant les principales logique de jeu
- └─Models //Répertoire contenant les éléments du jeu
- └─Board.cs //Fichier permettant d'initialiser le plateau de jeu
- └─Sprites //Répertoire contenant les images
- └─b.png //Photo pour les pions noirs
- └─w.png //Photo pour les pions blancs
- └─Form1.cs //Fichier contenant la logique de la partie graphique
- └─Form1.Designer.cs //Fichier contenant la partie graphique du jeu
- └─Program.cs //Fichier servant de point d'entrée et de lancement de jeu

### CheckGameTestUnitaire //Fichier permettant la création et l'initialisation des tests

- └─Dépendance //Bibliothèque externe ou interne
- └─Analyseurs //Permet de vérifier les bonnes pratiques/erreurs
- └─Frameworks //Répertoire où se trouve les Frameworks
- └─Packages //Regroupe les bibliothèques tel que NuGet
- └─Projets //Liste des sous projets ou module de la solution
- └─BoardTest.cs //Fichier servant à créer et faire les tests lié au plateau de jeu
- └─Form1Test.cs //Fichier servant à créer et faire les tests lié au jeu côté graphique
- └─GameControllerTest.cs //Fichier servant à créer et faire les tests lié au principale logiques du jeu

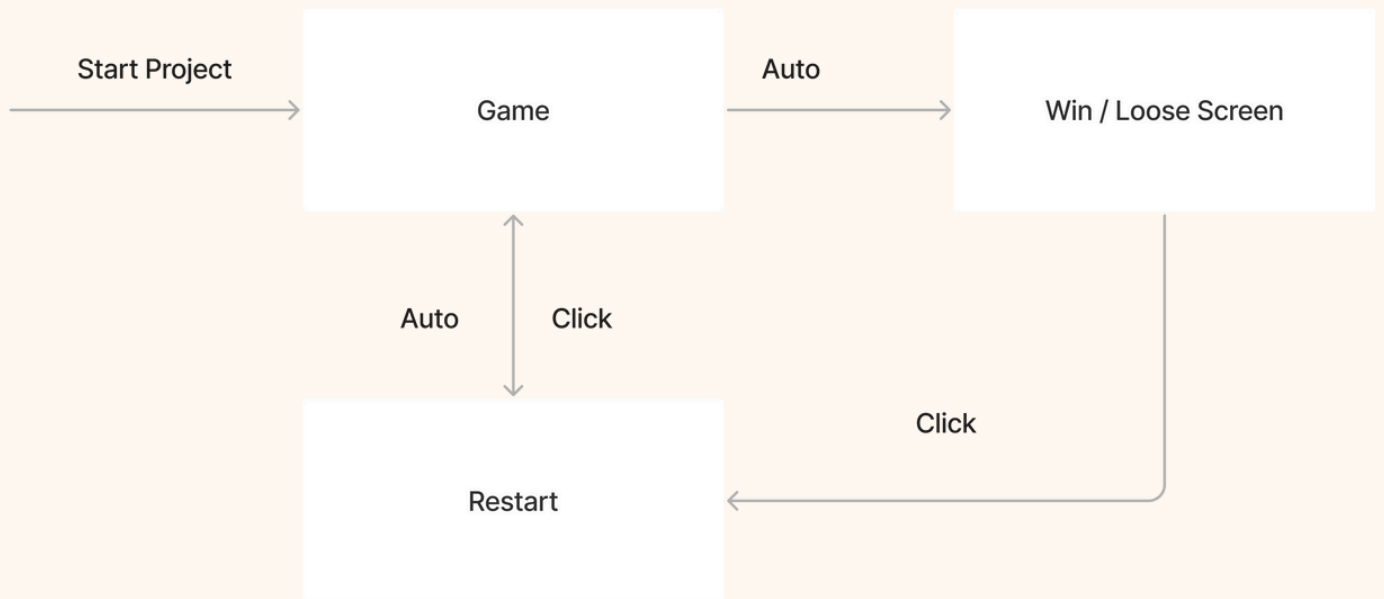
# Wireframe / Mock up

Game Wireframe





# User Structure



# API documentation

## Les documentations

- Pour les tests vous pourrez retrouver les détails sur le excel: *Plan de test.xlsx*
- Pour des détails sur la conception du jeu le cahier des charges peut aider: *Cahier des charges jeu de dame.pdf*

# Glossaire

## Définition des termes techniques utilisés dans la documentation technique

Technical Design: Conception technique de jeu vidéo, c'est le lien qui fait se joindre les principes de game design et les problématiques techniques

Project Architecture: Architecture de projet, c'est la façon et la manière par lequel le jeu a été élaboré

Project Structure: Structure de projet, c'est la l'arborescence du jeu ainsi que la définition et l'utilité de chaque fichier/dossier du projet

Wireframe/Mock Up: C'est un schéma "squelettique" en 2D qui permet de visualiser les début d'une page Web ou d'une application de même pour les jeux