

Documentation du Jeu de Dames

1. Introduction

Le jeu de dames développé est une application Unity où deux joueurs peuvent jouer ensemble sur un même écran, en tour par tour. Ce projet propose une expérience de jeu traditionnelle avec des éléments modernes tels que des animations simples, des messages d'interaction utilisateur, un menu simple et un plateau de jeu 8x8. Les joueurs alternent leurs tours, et on est libre d'appliquer la variante du jeu que l'on souhaite. La victoire est déclarée lorsqu'un joueur n'a plus de pions ou ne peut plus jouer.

2. Organisation du projet

Nous avons utilisé GitHub et discord comme moyen collaboratif pour l'échange d'informations et de codes.

Retrouvez-ici le lien vers notre page github : <https://github.com/Docteur-Jemper/Projects2024.git>

Nous avons créé une branche à notre nom que nous avons régulièrement fusionné avec la branche principale (master) du projet

2. Fonctionnalités Essentielles

- Les joueurs peuvent déplacer leurs pions sur des cases valides, selon les règles classiques du jeu de dames.
- Les pions adverses peuvent être capturés en sautant par-dessus
- Lorsqu'un pion atteint l'extrémité opposée du plateau, il est promu en dame.
- La partie se termine lorsqu'un joueur n'a plus de pions.
- Le jeu se déroule en tour par tour, le joueur devant lui-même valider la fin de son tour
- Les messages indiquent le joueur en cours, les erreurs potentiels en partie ou la victoire.

3. Fonctionnalités Additionnelles

Le ciel change aléatoirement au lancement du jeu et tourne lentement pendant la partie.

Les déplacements des pions déclenchent un effet sonore.

Les joueurs peuvent choisir un thème de plateau (classique ou bois).

Les joueurs peuvent revenir au menu principal depuis la scène de jeu.

4. Commandes et Raccourcis

- **Clic gauche (Souris) :**
 - Sélectionner un pion ou une case valide pour déplacer un pion.
- **Clic droit (Souris) :**
 - Supprimer un pion (pour des parties libres ou des tests).
- **Touche Entrée (Clavier) :**
 - Valider un tour et passer au joueur suivant.
- **Échap (Clavier) :**
 - Retourner au menu principal.

5. Structure du Code

1. **CheckersGame.cs :**
 - Gère les règles principales, l'alternance des tours, la gestion des pions et la détection des conditions de victoire.
 - Interagit avec l'interface utilisateur pour afficher les messages de progression et de fin de partie.
2. **PawnScript.cs :**

- Contrôle le comportement des pions, comme la mise en surbrillance, la suppression et la promotion en dame.

3. **PositionScript.cs** :

- Gère les interactions avec les cases du plateau, comme le déplacement des pions vers une case sélectionnée.

4. **RandomSky.cs** :

- Change aléatoirement le ciel au lancement du jeu et applique une animation à notre ciel pour une meilleure ambiance visuelle.

5. **UIScript.cs** :

- Gère les actions utilisateur dans le menu principal et les interactions comme le retour au menu principal.

6. **GameOptions.cs** :

- Permet de conserver les choix des joueurs (comme le style de plateau) entre les scènes.

Vous retrouvez ces fichiers dans le dossier **Assets/Scripts/**.

6. Tests Unitaires

Des tests NUnit sont implémentés pour vérifier les fonctionnalités clés :

- Changement de tour.
- Sélection, déplacement et suppression de pions.
- Conditions de victoire.

Vous retrouverez les versions de test des fichiers C# précédemment présenté dans le dossier **Assets/Tests/**.

7. Interface Utilisateur

5.1 Menu Principal

- **Boutons** :

- **Jouer** : Lance la partie et charge la scène de jeu.
- **Quitter** : Ferme l'application.
- **Sélection Plateau** : Permet de choisir entre le plateau classique ou en bois.

5.2 Scène de Jeu

- **Affichage** :
 - Indication du joueur actuel.
 - Messages d'erreur en cas de mauvais déplacement.
 - Message de fin de partie avec le gagnant.
- **Éléments interactifs** :
 - Les pions et les cases du plateau.

8. Fonctionnement du jeu

Lancement :

- Le joueur arrive sur le menu principal.
- Il décide soit de lancer une partie, soit de modifier les paramètres, soit de quitter le jeu
- Un ciel aléatoire est affiché en arrière-plan au lancement d'une partie

Sélection du plateau :

- Les joueurs peuvent choisir un style de plateau.
- Une fois sélectionné, le choix est appliqué dans la scène de jeu.

Déroulement de la partie :

- Les joueurs jouent en alternance :
 - Sélection d'un pion valide (clic gauche)
 - Déplacement vers une case valide ou capture d'un pion adverse (clic gauche sur la nouvelle case)

- Un joueur peut manger un ou plusieurs pions adverses (clic droit sur le ou les pions mangés pour les faire disparaître)
- Une fois un déplacement effectué, la touche Entrée passe au tour suivant.

Promotion en dame :

- Lorsqu'un pion atteint l'extrémité opposée, il est promu en dame (avec une couronne affichée).

Fin de partie :

- Si un joueur n'a plus de pions, un message de victoire s'affiche.
- Après 2 secondes, les joueurs sont redirigés vers le menu principal.

9. Organisation des dossiers

Assets/Fonts/ : Contient les polices téléchargées

Assets/Images/ : Contient les images importées

Assets/Imports/ : Contient les modèles 3D et actifs téléchargés en provenance de l'Unity Asset Store et d'autres sites externes

Assets/Prefabs/ : Contient les préfabriqués des pions et des cases.

Assets/Scenes/ : Contient les différentes scènes de jeu (scène du menu et scène de jeu)

Assets/Scripts/ : Contient tous les scripts C#.

Assets/Sounds/ : Contient les clips audio.

Assets/Tests/ : Contient les fichiers de tests unitaires en C#

Les autres dossiers sont des dossiers de configuration et n'ont pas d'importance dans le cadre de notre projet.

10. Points d'amélioration

Comme rien n'est parfait, nous avons pensé à quelques points d'amélioration à la suite de ce projet :

- Implémentation une IA pour des parties solo
- Permettre d'enregistrer des parties
- Ajouter en option la possibilité de sauvegarder et reprendre une partie
- Ajouter une fonctionnalité multijoueur via réseau

11. Informations et documents en annexe

Ce projet a été réalisé dans le cadre du cours de Qualité de Développement par :

- Antoine PISSON
- Matthias BILLA--TECHER
- Tom FLETER
- Gabriel HESDIN

Membres du groupe BUT 3 Dev2.

Retrouvez ci-joint le lien vers le dépôt du projet <https://github.com/Docteur-Jemper/Projects2024.git>

Vous retrouvez l'intégralité du projet réalisé sous la version 6000.0.32f1 d'Unity.

Un exécutable sera également en annexe avec ce document.

Cordialement,

Antoine PISSON