

Application Qt : échec



Table of Contents

Présentation.....3

Instructions.....3

Fonctionnalités.....3

Description technique.....5

Présentation

Le projet consiste en un jeu d'échecs développé en C++ avec l'utilisation du framework Qt. Il offre une interface graphique interactive, permettant aux joueurs d'interagir avec le jeu à l'aide de la souris pour sélectionner et déplacer les pièces sur le plateau de jeu.

Instructions

Étapes pour compiler et exécuter le programme :

1. Assurez-vous d'avoir installé Qt version 5.15.3 (ou supérieur) sur votre système.
2. Clonez le dépôt du projet depuis <https://github.com/ThomasBouvais/chess>.
3. Ouvrez le projet dans Qt Creator.
4. Configurez les options de compilation selon votre système.
5. Compilez le projet.
6. Exécutez l'application.

Aucune dépendance n'est requise.

Fonctionnalités

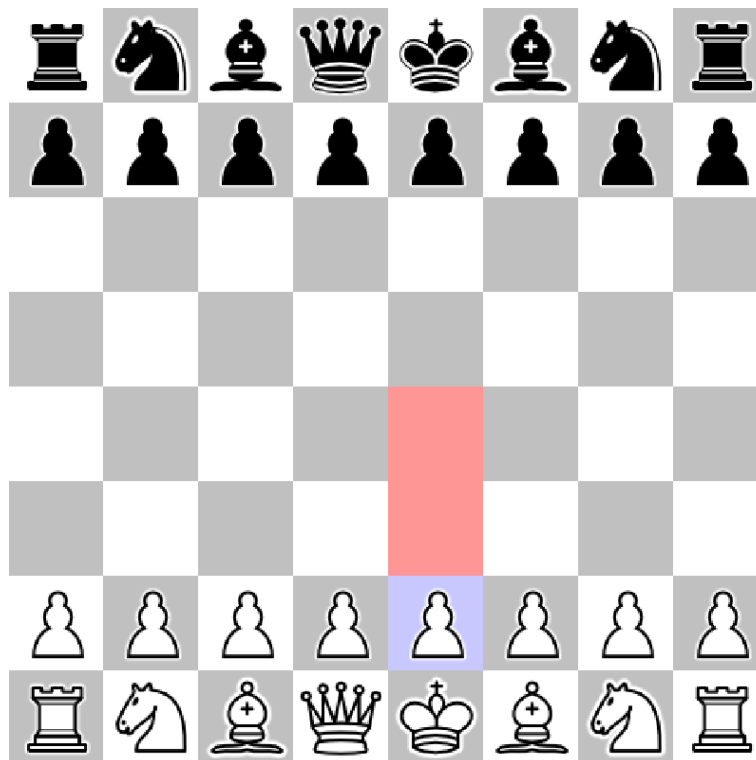
Déplacement de base pour chacune des pièces : Les pièces se déplacent conformément aux règles classiques des échecs. Chaque type de pièce a ses propres règles de déplacement.

promotion des pions : Lorsqu'un pion atteint la dernière rangée du plateau adverse, il est automatiquement promu en une reine.

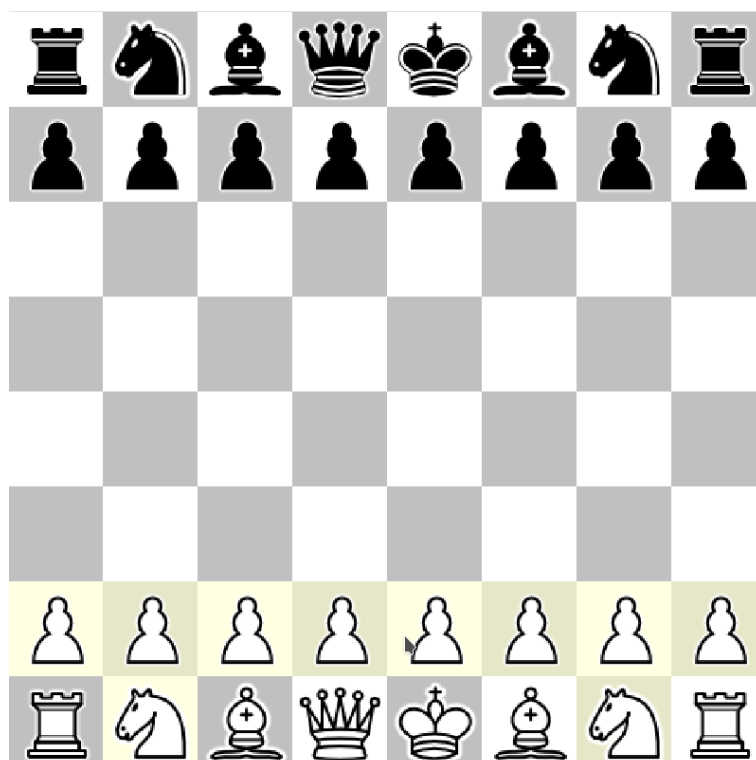
Détection de mise en échec : Le jeu détecte automatiquement lorsque le roi d'un joueur est menacé (en échec), informant ainsi le joueur de la nécessité de protéger son roi.

Détection de fin de partie : Le jeu détecte automatiquement les conditions de fin de partie, telles que l'échec et mat ou le pat. Lorsqu'une partie se termine, une boîte de dialogue s'affiche pour annoncer le résultat de la partie.

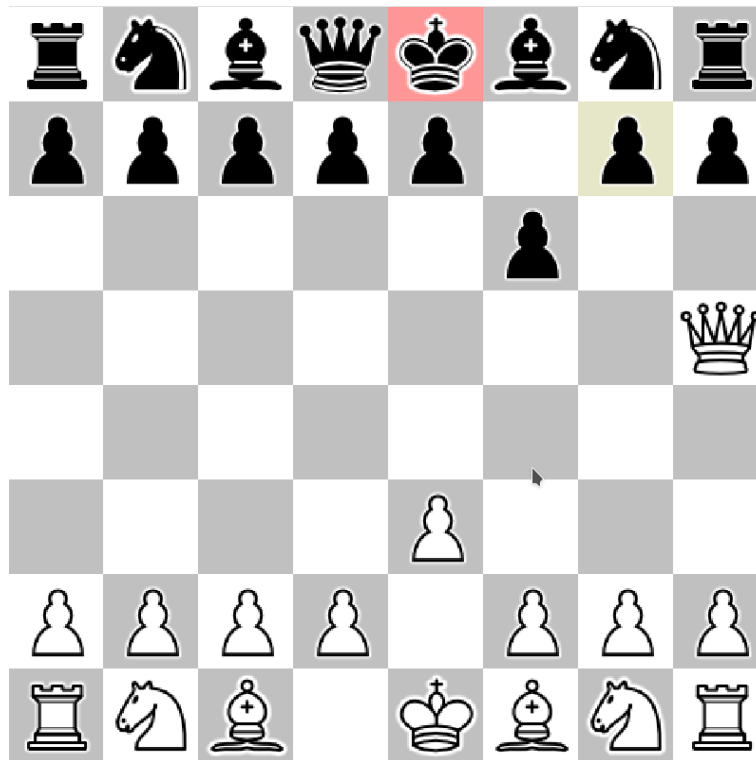
Interface graphique interactive : Pour interagir avec le plateau, le joueur clique d'abord sur la pièce qu'il veut déplacer puis sur la case cible. Lorsqu'une pièce est sélectionnée, elle et toutes les cases qui sont disponibles pour la déplacer sont mises en surbrillance.



Lorsque c'est le tour d'un joueur, toutes ses pièces qui sont jouables sont mises en surbrillance.



Quand le roi est en échec, il est mis en surbrillance, seul les déplacements qui empêchent cet échec sont disponibles (et s'il n'y a pas d'échec, tous les coups qui résultent en un échec sont également retirés)



Écran de victoire :



Description technique

Utilisation de la Programmation Orientée Objet (POO) : Les pièces d'échecs ont été implémentées en tant qu'objets distincts avec des comportements spécifiques pour chaque type de pièce.

Utilisation de QPoint et QVector : Utilisés pour la gestion du plateau de jeu et des différentes coordonnées.

Utilisation de QPainter pour dessiner l'interface utilisateur : La classe QPainter a été utilisée dans la méthode pour dessiner le plateau de jeu et les pièces. Les différentes fonctions de QPainter ont été exploitées pour dessiner les cases du plateau, mettre en surbrillance les cases sélectionnées et afficher les pièces.

Gestion des événements de souris avec QMouseEvent : La classe QMouseEvent a été utilisée pour détecter les clics de souris de l'utilisateur sur le plateau de jeu. Cela a permis de sélectionner et de déplacer les pièces en réponse aux actions de l'utilisateur.

Création de boîtes de dialogue personnalisées avec QDialog et QVBoxLayout : Les classes QDialog et QVBoxLayout ont été utilisées pour créer des boîtes de dialogue personnalisées pour afficher les résultats de la partie. Ces boîtes de dialogue ont été utilisées pour afficher des messages de victoire ou de match nul à l'utilisateur à la fin de la partie.

Difficultés rencontrées et améliorations possibles :

J'ai eu quelques difficultés pour implémenter la fonction pour vérifier que le roi n'est pas en échec notamment pour vérifier qu'un roi ne met pas en échec un autre roi, la fonction de déplacement des pions adverses étant appelées pour enlever tous les coups qui résultent en un échec cela créait une boucle infinie.

Pour améliorer le jeu, il faudrait rajouter les quelques règles avancées manquantes : le roque, la capture « en passant », lorsqu'un pion arrive au bout du plateau il ne peut être transformé qu'en reine, le jeu peut détecter une égalité par un pat mais pas par répétition. L'écran de victoire à la fin pourrait également être amélioré pour être plus joli.