

Design Document for Checkers Project

by Jiaao He, 2016011279

Introduction

Purpose

Write a QT gui, network module and a Checkers core to implement the logics of a game.

Scope

Programming training

System overview

A server is set up as the user clicks. Then two users connect to the server to start the game. A gameboard is generated and users can perform their operations until the end of the game. The board will sync automatically when a player performs a move.

System Architectural Design

Architectural Design

This project is divided into three parts.

1. The logic part of a checkers game, which judges which move the player can perform.
2. The server class which provides network communicating service and determines the turns.
3. The client which displays the board and collect the players' actions.

Decomposition Design

Board class

This class judges if a piece is movable from one grid to another, provided the whole board. In order to detect the longest continuously jumping route, *Depth First Search* tech is used, which performs fast enough to support real-time judging due to the small scale of the relatively board.

Server class

This class is based on `QTcpServer` class, which waits for two client to establish `QTcpSocket` connection. And then it work as a pipe while during the board syncing work, and do some logic work when a player wants to admit defeat.

MainWnd class

This class controls the display window of a board. A specific widget is deployed to display the board, as its paint event is redirected.

Button press events and board settings are also implemented here. The server class is stored as a member of this class, in convenience of controlling and restarting.

Data Design

Data Description

For a piece on the board, the lowest two bits are used to record its faction. The third bit is used to record if it is a king. The forth bit is used to record if it is an obstacle as a continous jump is performing.

Human Interface Design

Overview

Paint manually on a widget and collect its mouse events to do actions.

Screen Images

Skipped.