**课程实践报告**

1. 内容简介

本次课程实践选择的是第二个方式，具体内容是制作一个能在Windows系统上运行的ChessGUI（名称为SimpleChessGUI）。

ChessGUI 是一种图形用户界面 (GUI) 应用程序，专门用于国际象棋游戏。它提供了一种直观的方式来进行国际象棋对局，无论是人对人、人与计算机，还是计算机对计算机。

ChessGUI 的主要功能包括：

1. 棋盘和棋子显示：一个清晰的棋盘界面，允许用户直观地看到和移动棋子。

2. 棋局保存和加载：用户可以保存当前棋局并在以后加载继续。

3. 棋谱记录：记录并显示每一步棋的棋谱，方便回顾和分析。

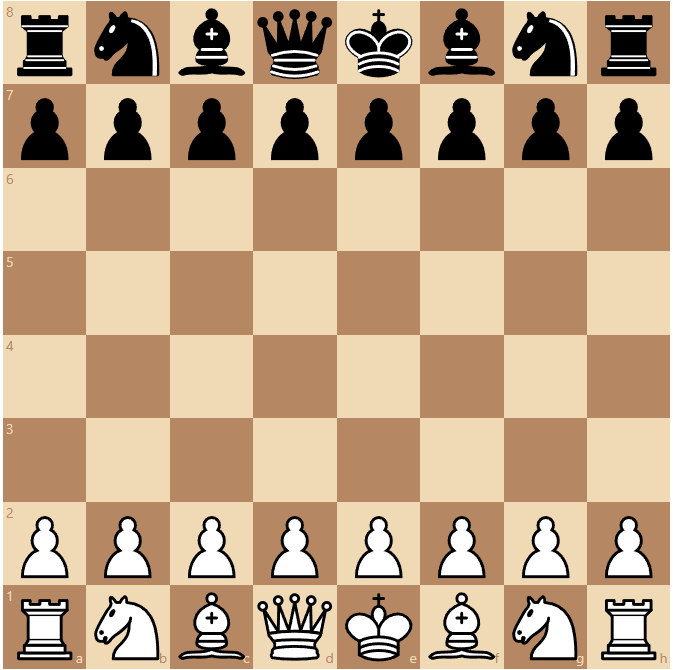
4. 与棋力引擎的集成：支持连接多个国际象棋引擎，如 Stockfish，以便进行人机对战或分析棋局。

1. 软件的主要功能

SimpleChessGUI的主要功能包括以下几个：

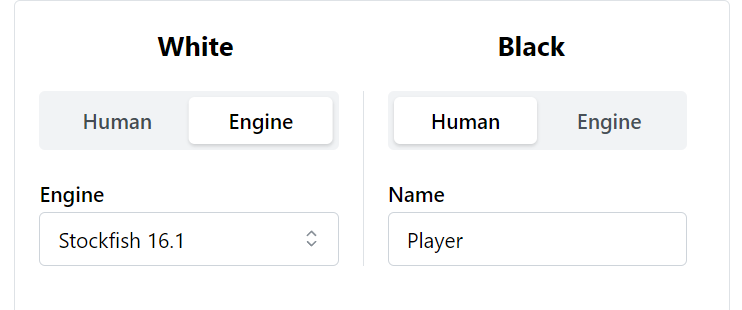
1. 提供棋盘显示。

用户可以通过鼠标拖动棋子下棋。在棋局开始前，或者当前回合并没有轮到用户时，棋子将不可拖动。



1. 提供人对人，人与计算机和计算机对计算机的棋局。

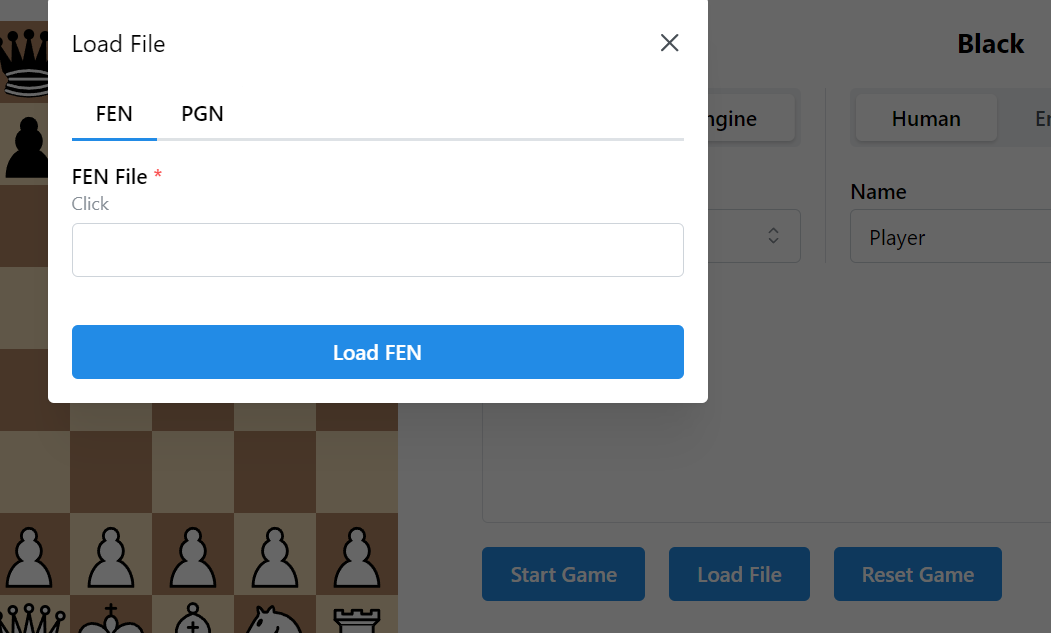
通过选择Human和Engine滑块，设置黑、白方分别是由人类还是计算机参与。如果是设定为人类，需要输入棋手的名称；如果设定为计算机，则需要在预先加载的引擎中选择。



1. 提供预先加载棋局的功能。

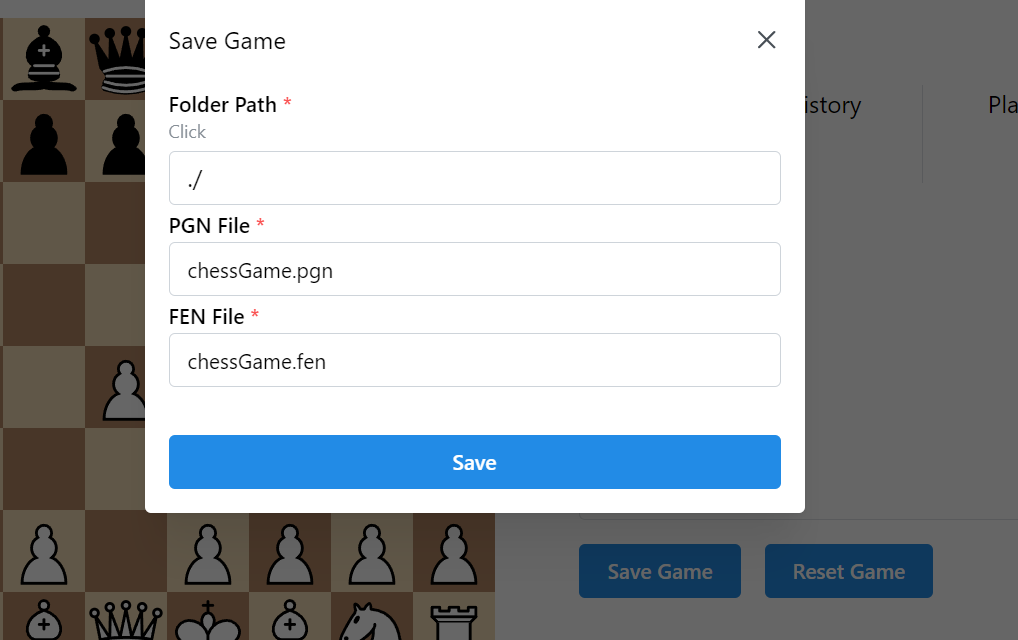
点击Load File按键，可以选择加载.fen或.pgn格式的文件（FEN 是一种用于描述国际象棋局面的字符串格式。它能记录一个棋局的具体状态，包括棋盘上的所有棋子的位置、哪一方走棋、是否可以王车易位、可被吃过路兵的位置、半回合计数和全回合计数，PGN 是一种用于记录国际象棋对局的标准棋谱格式。它不仅记录了每一步棋的走法，还可以包含对局的元数据，如对局双方的名字、比赛日期、地点、结果等）。

如果加载的fen或pgn文件存在错误，则点击Load FEN（或Load PGN）后不会加载。成功加载后，棋盘会立即设置为文件中记录的棋局，此时用户可继续设置黑、白方的棋手。



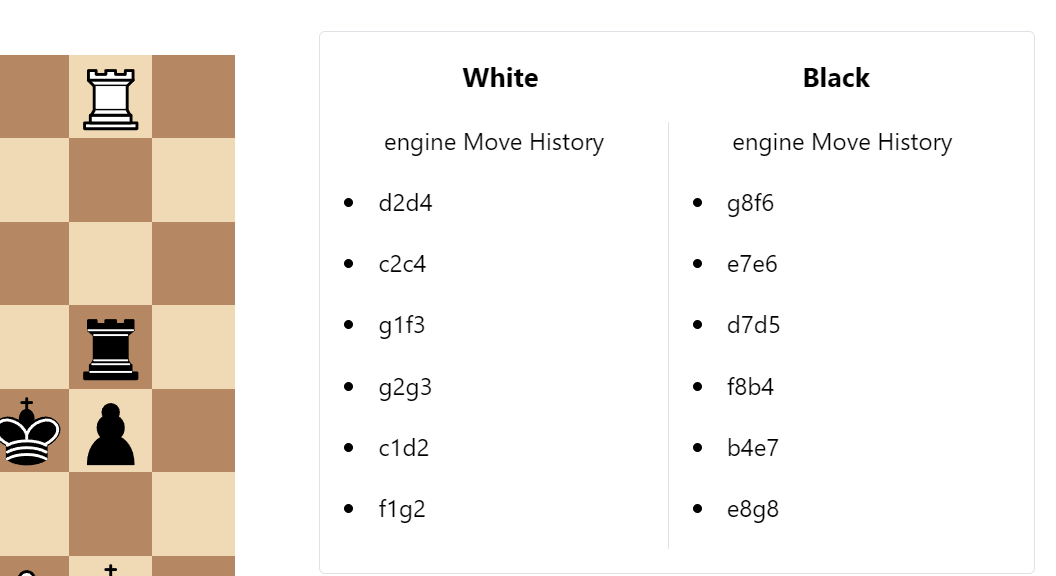
1. 提供保存棋局的功能。

点击Start Game开始下棋后，点击Save按键会跳出保存棋局的弹窗，通过弹窗选择保存文件的文件夹路径和.fen与.pgn文件的文件名。点击Save按键后，.fen和.pgn文件会被保存到指定的文件夹内。



1. 提供棋谱记录的功能。

采用简化代数记谱法在棋盘右侧记录双方棋手的落子记录，通过移动的起点和终点位置来表示走法。



1. 提供棋局分析功能。

在参与下棋的一方为象棋引擎时，点击Show Analysis，会在棋盘右下方显示针对当前棋局的分析，输出当前棋局的Score（以cp为单位）和PV。

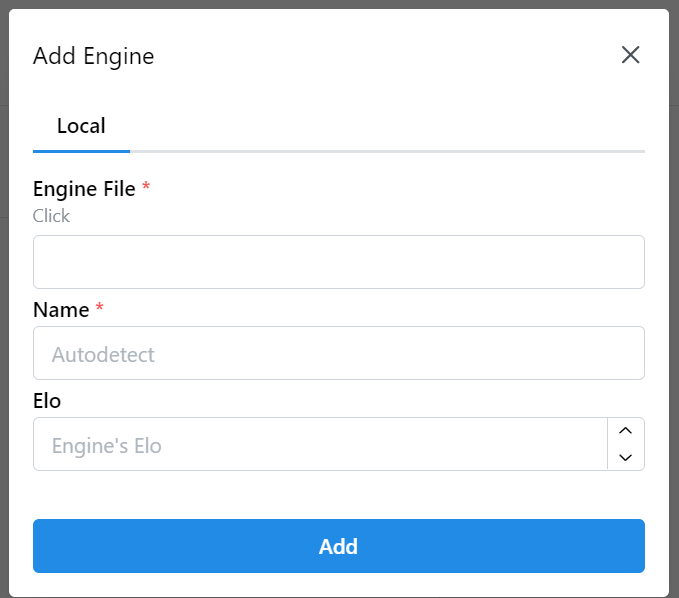
Score表示当前局面进行评估后得出的分数，用于表示一方的优势。正值表示白方优势，负值表示黑方优势。cp是“千分之一步”，即1/100的兵值。例如，+50 cp 表示白方的优势相当于半个兵的价值，而 -200 cp 表示黑方的优势相当于两个兵的价值。

PV表示主变着（Principal Variation），即引擎在分析当前局面时认为的最佳走法序列。这个序列通常是引擎预测的最优解，假设双方都按照最优走法进行。



1. 提供导入引擎的功能。

在引擎界面，点击Add Engine，会跳出加载引擎的选项。用户需要指定引擎的可执行文件的路径。引擎名称可以由用户手动输入，也可以在添加可执行文件时由软件自行设定。



1. 提供设置引擎参数的功能。

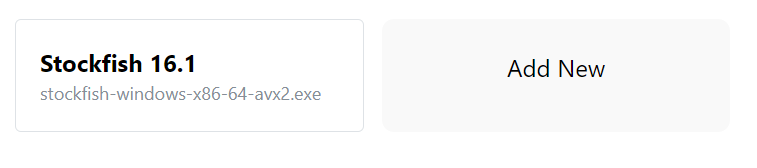
在引擎界面会以卡片形式显示已添加的引擎。点击卡片会显示该引擎的设置界面。

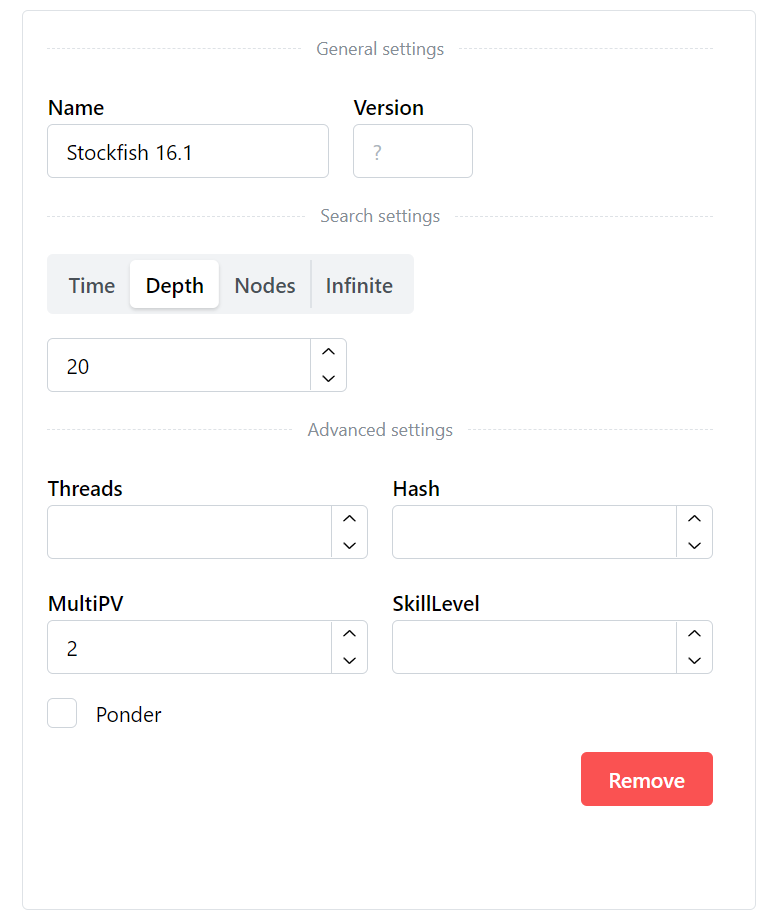
在General settings部分，可以设置引擎的名称和版本。

在Search settings部分，可以设置引擎停止搜索的条件。选择Time, Depth, Nodes分别表示在搜索时间、搜索深度和搜索节点上限制引擎搜索的范围，选择Infinite则代表引擎会一直搜索，不会停止。

在Advanced settings部分可以设置引擎运行的线程数、哈希表大小、返回的最有结果数、引擎技巧级别以及是否可以在对手思考时搜索。

点击Remove按键则会将该引擎从列表中移除。





1. 软件的主要结构

前后端均由JavaScript实现，以React+Electron作为框架，并借助Vite实现前端的开发和构建。下面从前端和后端介绍软件的主要构成部分。

前端：

前端通过React实现，主要是由各种自定义的React组件构成的，以下介绍主要的自定义组件。

1. App组件

所有其他前端组件的父组件，其中包括两个路由，分别指向Board组件和Engine组件，即棋盘界面和引擎界面。

1. Board组件

呈现棋盘界面的组件，其中包括SaveModal，LoadModal，ChessGame，OpponentForm等组件，分别实现保存弹窗、加载弹窗、棋盘和棋手设置的界面。棋局分析的界面没有使用自定义的组件来实现，而是直接在Board组件中实现。

1. SaveModal组件

呈现保存棋局的弹窗界面，用户在这界面可以设置保存棋局文件的路径。

1. LoadModal组件

呈现加载棋局的弹窗，向用户提供加载文件的UI，并在加载出错时弹窗提示。

1. ChessGame组件

呈现棋盘，并在下棋的过程中与处理用户的交互，同时通过UCI（通用国际象棋接口）和用户添加的引擎交互。

1. OpponentForm组件

在准备阶段（点击StartGame前）向用户提供设置双方棋手的UI，在下棋阶段显示双方棋手的走法记录。

1. Engine组件

呈现引擎列表界面，显示用户已经添加的引擎，并提供添加引擎的UI。包括AddEngine，EngineCard，EngineSettings组件。

1. AddEngine组件

呈现添加引擎的弹窗，用户在此界面设定可执行文件路径后，组件将通过UCI确认可执行文件是否是国际象棋引擎，若不是则不会添加。

1. EngineCard组件

呈现用户已添加的引擎，包括引擎名称和可执行文件路径名。

1. EngineSettings组件

在用户点击EngineCard时，显示相应的引擎的详细设置界面，同时提供移除引擎的UI。

后端：

后端主要包括以下几个部分：

1. chessFile.js

读取棋局文件，包括.fen和.pgn格式的文件。

1. engine.js

与引擎相关的后端处理，包括读取引擎信息、保存引擎、删除引擎、加载引擎、向引擎请求当前局面的最佳走法和局势分析等。

1. saveGame.js

将当前棋局的信息以.fen和.pgn文件保存。

‘

1. ipcHandlers.js

后端注册的监听器，用于监听前端的请求，并将请求的结果返回给前端。

1. main.cjs

程序的入口文件。