Problem Statement and Goals
Chess Connect

Team #4,
Alexander Van Kralingen, vankraa
Arshdeep Aujla, aujlaa4
Jonathan Cels, celsj
Joshua Chapman, chapmj3
Rupinder Nagra, nagrar5

2022-09-22

Table 1: Revision History

Date	Developer(s)	Change
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

1 Problem Statement

Online chess has functionality for both beginners and experienced players to learn and practice the game. However, these forms of learning place emphasis on a visual style of learning using a standard keyboard and mouse, while physical boards place emphasis on tactile learning when learning or studying the game. The highest rated chess players often use a combination of the two styles to optimize their play. However, no option exists for players of any skill level to integrate their over-the-board and online play with one solution. This project plans to centralize these two mediums of studying the game in order to provide flexibility and remove constraints from new players in learning how to play chess.

1.1 Problem

1.2 Inputs and Outputs

[Characterize the problem in terms of "high level" inputs and outputs. Use abstraction so that you can avoid details. —SS]

1

1.3 Stakeholders

This project will be able to assist chess players of any level looking for a tool to help them learn and study the game. For beginners, the board serves as a learning tutorial and a general introduction to the game, while intermediate and advanced players can use the machine learning move recommendations to study new lines, puzzles, and specific positions for practice. In addition to chess enthusiasts, this project will also be relevant to chess tournament organizers looking for a method to easily broadcast and share their games online in real time.

1.4 Environment

Due to the web application component of our project, many of our languages will revolve around a standard full-stack development tech stack. Because of this, we will be using JavaScript, HTML, CSS, and Python. React is a front-end framework that encompasses JavaScript, HTML, and CSS in a single framework. We are also planning to use Node and FastAPI for our back-end which both make use of Python. The C programming language will be used as a middleware to transmit information from the hardware to the web application. To deploy our web application online we will be using a cloud application platform called Heroku, and testing will be done using PyTest and the React Testing Library which is part of the React framework. In order to store previous games, we might also make use of databases such as MySQL and MongoDB, relating to some of our tasks in the stretch goals section below. In terms of the hardware, we are planning to use — for the micro-controller and — for the display, where our method of connecting the hardware and software together will be Bluetooth. Our documentation will be done using Overleaf, an online LaTeX editor which will be connected to our personal Github repository. We will also be using Thode Makerspace as a workshop to build our project.

2 Goals

2.1 Live Broadcasting

Physical chess boards lack the ability to share and store moves with anyone outside of the room. Chess Connect will live broadcast the state of the game to a server and store the moves to share with others. This allows for players to communicate world-wide while maintaining the benefits of a physical board.

2.2 Engine Integration

Online chess has taken large leaps in recent years due to chess engine development and implementation growing immensely. Physical boards lack this real time evaluation and potential for improvement. Chess Connect will fetch best moves from an existing engine and display the moves on-board via LCD display.

2.3 Beginner Mode

For new players, nothing beats learning on a real board. It increases

3 Stretch Goals

3.1 Database

Providing users the ability to study their past games, allows them learn from their mistakes and blunders in the game and allow them to evaluate their own positions with a better understanding on how to play in future games. Using MySQL and MongoDB to store those games in these databases, we can let users look through all games previously played on our board.

3.2 Study Mode

Study Mode will allow users to create puzzles or positions that do not start from a default starting position of the game, instead allowing users to practice a specific phase of the game and the ability to take back moves without having to play out a full game with an opponent. This mode is meant to be used individually, also allowing a user to use and study the board without the necessity of another player.