Project Proposal
Checkmate

Abstract

The software project is a website that provides users with a platform to attempt various chess puzzles in a gamified way. Users are presented with a virtual, standard 8x8 chessboard in which they are prompted to solve a particular chess position starting from relatively easy to progressively becoming more difficult as each puzzle is solved. For each puzzle completed correctly, one point will be added to their counter. The user must find the best move on the prompted board to move on to the next puzzle. After three incorrect attempts are made, the user's progress resets.

The user will be initially told which color's move it is. If the puzzle says "Black to move" or "White to move" then that is the color that the user must move with. In addition to this, in order to initialize the position, the first move is automatically executed to show the user what was the last move played by the opposite color. The puzzles will require the user to "solve" the chess game by getting their color to win by checkmate in the allowed number of moves. Puzzles will be given randomly and progressively increase in difficulty. The user's scores will be saved.

Group Members

Trentin Barnhart

Autumn Hale

Esai Jimenez

Description

Chess is a classical board game that has been played for centuries. In recent years, there has been a surge in popularity in chess that introduced many new users to the game. Many beginners are reluctant to immediately jump right into the action with another player without gaining a little bit of experience first. Therefore, this software project is a website that provides users with a platform to attempt various chess puzzles in a digestible and interactive way. The website will contain a 2D 8x8 chessboard and presents users with a particular chess position, starting with relatively easy puzzles and progressively becoming more difficult as each puzzle is solved. Users are prompted to solve each puzzle by finding the best move in the given position. For each puzzle completed correctly, one point is added to the user's counter and the difficulty of the puzzle increases. If the user makes a single mistake on a puzzle, they are immediately taken to another puzzle around the same difficulty level as the previous puzzle they got incorrect. However, if the user makes three incorrect attempts in total, their progress is reset. When their progress is reset, the counter resets back to zero when they start the trial again. This encourages users to think critically and strategize before making their move.

The website will include a feature that provides a tutorial on how the chess pieces move if the user is unfamiliar with the mechanics of the pieces. This is an optional tool at the user's disposal in the event that they need help. There will be many users with a wide range of skill levels, therefore it is important to cater to both beginners and masters of the game.

As an extra layer of guidance, whenever the user taps on a piece they would like to move, there will be a series of dots that show the available legal moves the user can make with that particular piece. This will be a feature that the user can toggle on or off if they deem it necessary. Like mentioned earlier, there has been a major increase in popularity in chess, therefore there are

many newer players to the game. By adding this slight aid, it will allow users to not be discouraged to play due to them forgetting how the pieces move.

Additionally, the user is initially told which color's move it is. If the puzzle says "Black to move" or "White to move," then that is the color that the user must move with. In addition to this, to help the user understand the current position, the first move is automatically executed to show the user what the last move played by the opposite color. This allows the user to understand the context of the position and visualize the last action performed. The user will be challenged to "solve" the chess position by getting their color to win by checkmate in the allowed number of moves. This added goal encourages users to think critically and plan ahead to achieve checkmate.

Overall, this project provides a fun and interactive way for users to improve their chess skills. The use of increasingly difficult, random puzzles matched with a user-friendly interface offers a unique and challenging experience for users. This project is an ideal tool for chess enthusiasts of all skill levels.

Feature List

Minimum Viable Product	Time Permitted	Wishlist of Features (Phase 2)
2D, 8x8 chessboard graphical display - Website hosted on Firebase - Create and destroy chess pieces - Score counter - Attempts left counter - Move pieces if move is legal - Generate or read a pre-existing board state - Output board state to the visual display - Progress reset if three incorrect attempts are made - Tutorial on mechanics of chess pieces - Inform user which color must move - Difficulty rating - Dots to show legal moves when a piece is clicked on - Toggle for dots to show when a piece is clicked	- Leaderboard - Login - Timer/Clock - Option to view solution	- Themes/Backgrounds - Puzzle races against others simultaneously for most puzzles completed in one trial - Graph that shows progress - Flip board GUI - Profile page - List of all moves made so far - Sound when pieces move - Various modes - Additional chess related pages on website - Custom puzzles

Technology

- Platform
  - Google Chrome (most recent version)

- Operating System
  - Windows 10 and 11

- IDE
  - Visual Studio Code

- Programming Languages
  - Python
  - HTML
  - CSS
  - JavaScript

- 3rd party libraries and tools
  - Lichess Open Database
  - Figma
  - Python Chess Library
  - Python Pandas Library
  - Git
  - Github

- Server software
  - Google Firebase

- Communication software
  - Discord
  - Cell Phone as backup

Server Information

We intend on utilizing Google Firebase for its various features such as for authenticating users and hosting our website.

Data Sources

- Lichess.org open database
  - https://database.lichess.org/#puzzles

Team Members' Backgrounds

Autumn Hale is familiar with all languages chosen and listed on this document. She is also familiar with Windows, Visual Studio Code, Google Chrome, and Discord. She is not familiar with Google Firebase. She knows how to play chess and has solved some chess puzzles. She has not used the data source API before. Her main responsibilities will be the frontend.

Trentin Barnhart is familiar with Java and is comfortable with the rules of chess and the concept of the project. He is not very familiar with web based development by comparison. He is comfortable with the technology being used (VSCode, etc.) but will have to learn more web based skills to help with the online portions. Main responsibilities will be the backend.

Esai Jimenez is familiar with Windows, Visual Studio Code, Google Chrome, and Discord. He has a high understanding of chess and chess puzzles. In addition, he has some experience with working with HTML, JavaScript, and CSS. Very little experience with databases and Python. Main responsibilities will be dealing with hosting the website and database, while helping when needed with the backend and frontend.

Dependencies, Limitations, and Risks

One primary dependency, and therefore major risk, is Lichess' Open Database that will be utilized for Checkmate's puzzles. This resource may not always be available for free use under Creative Commons CC0 license. The resource may also give corrupted data, incorrect and/or inaccurate data, or no data at all for whatever reason. This may be solved by an error message on Checkmate's website. Team members should be prepared to look into backup sources that will provide the same information in order to ensure that Checkmate runs regardless if this source fails or not.

Another primary dependency and risk is Google's Firebase. This platform may no longer fit our needs at some point during the project's life cycle, and therefore may either need to be changed or upgraded in some way. No team members have used the platform before, and therefore more time and effort will need to be spent in this area in order to learn and ensure project success. This may have repercussions that reverberate to other areas of the software project.

Timeline

Week of:	Esai	Trentin	Autumn
January 22nd - January 28th	Learn about Google Firebase and figure out how to use the Lichess puzzle file.	Become familiar with technology being used and begin next step early.	Get an understanding for technologies that will be used and how they will work together.
January 29nd - February 4th	Implement what I learned about Firebase to host our website and utilize the information in the Lichess puzzle file.	Figure out what to use to get a visual display of the chessboard.	Create template mockup of the website and collaborate to ensure that it meets MVP.
February 5th - February 11th	Help Trentin and/or Autumn with any issues or facilitate their coding.	Work on the visual display so that there is something solid for the other two to use.	Code the bare front end of the website according to mockup.
February 12th - February 18th	Collaborate with Autumn to finish the 2D, 8x8 chessboard interface and have it be hosted on Firebase.	Work on control functionality for a while (I anticipate this will take me some time).	Chess board interface put on the website. Collaborate on hosting the website.
February 19th - February 25th	Collaborate with Trentin to finish the functionality of the chess pieces.	Collaborate with Esai to finish the chess board functionality.	Chess board piece interpretation from data to basic GUI onto the board. Graphical fixes and other frontend work.

February 26th - March 4th	Collaborate with Trentin on generating a random puzzle position.	Collaborate With Esai on generating random puzzle positions from the database.	Chess board piece interpretation from data to basic GUI onto the board. Graphical fixes and other frontend work.
March 5th - March 11th	Spring Break March 4-12.	Spring Break March 4-12.	Spring Break March 4-12.
March 12th - March 18th	Work with Autumn in adding the visual features to the interface to make it look professional.	Collaborate with the group to connect our backend and frontend.	Collaborate with Esai and Trentin to connect the backend and frontend. Ensure the mechanics of the pieces work with the interface.
March 19th - March 25th	Work with Trentin to get the functionality for the tutorial.	Work with Esai to get tutorial functionality.	Ensure the mechanics of the pieces work with the interface.
March 26th - April 1st	Try to add a login feature and leaderboard if time allows.	Cleaning up bugs, doing some basic polish, and filling any gaps we left so that we have an MVP ready.	Ensure the mechanics of the pieces work with the interface. Graphical fixes and other frontend work.
April 2nd - April 8th	Collaborate with Autumn to ensure the mechanics of the pieces work with the interface.	Cleaning up bugs, doing some basic polish, and filling any gaps we left so that we have an MVP ready.	Graphical fixes and other frontend work. Clean up and bug fixes.
April 9th - April 15th	Work with both group members to ensure that we have a minimum viable product for the upcoming presentation.	Work with the group to ensure our minimum viable product for the upcoming presentation.	Work with both group members to ensure that we have a minimum viable product for the upcoming presentation. Clean up and bug fixes.

April 16th - April 22nd	Ensure there's a workable demo to show our vision at the Innovation Experience.	Ensure there's a workable demo to show at the Innovation Experience.	Ensure there's a workable demo to show our vision at the Innovation Experience. Clean up and bug fixes.
April 23rd - April 29th	Make changes needed for the final presentation.	Final Presentation Changes.	Make changes needed for the final presentation. Clean up and bug fixes.
April 30th - May 6th	Make finishing touches and present our final project.	Finishing Touches and presenting the final project.	Make finishing touches and present the final project.