

A4: Interactive Visualization Writeup

For our interactive visualization, we wanted to explore the open source lichess.org database, which keeps data of all of its games. The main thing we wanted to explore were chess openings. Chess is an extremely complex and expanding game, whereas after several moves, virtually every game is unique, so we decided to focus on the opening and see if there were any trends or interesting observations that could be seen about different opening moves and their success rate across chess rating (a metric of Elo used by lichess). Using a sample from the most recent games from lichess, we were able to create such a visualization.

Being an interactive visualization, we thought it would be super interesting and engaging to the user if they could actually see the board and play through a game of chess, so we spent part of our time integrating JS libraries to create a chess board. Matthew spent about 3 hours exploring chessboard.js and chess.js and integrating it into the site, whereas one could play through a game of chess and create a PGN, or Portable Game Notation, which is a standard way to record the series of moves made. This PGN then gets sent to our actual visualization to filter by, as the dataset of games has the PGN for the entire game. This way the user could make moves on the board and see info about their opening/move-set in real time, as the PGN for graph updates with every move! Another feature that we thought would be important and improve the user experience would be to add buttons that make moves for you. One is a reset button that would reset the board and the PGN that gets sent to the graph. The other is one that will play the next most common move based off of the dataset. It looks through the set or subset of the data being used by the graph and looks at each PGN, parsing and finding the next move in order to find the next most common move so that, when the button is pressed, it will automatically play that move, updating the board and subsequently, the graph. Matthew also added a status string on top of the board to help guide the user in what they can do, such as prompting them to make a move or informing them when there's no common move or no games that have the given opening. These additional features (along with some styling) took an additional 2 or so hours for Matthew. Overall, we wanted the users to be able to effectively visualize and have fun with our chess-based visualization, so our main way of user-interaction was through a chess board itself, along with some additional data that could hopefully help them learn how to use it and indicate what was happening every time they made a move!

We chose a stacked bar chart to show the success rate of each opening, because both wins and draws are significant outcomes for the chess player. In fact, most chess games at the highest levels end in a draw. In the bar chart, we represent wins in green and draws in blue, which have positive and neutral connotations, respectively. To show how the win/draw rates change as the player makes a move, we animate the graph with transitions to show the direction of change in the win and draw rates. This way, the user can easily see whether a particular move they make results in a better or worse outcome across ratings. We also chose to bin the data points by rating in increments of 100, as we thought that it would be a sufficient

interval to separate the data by rating without cluttering up the visualization. Although the bar chart shows percentages, as we care more about the rate of success, the actual number of games in each bin is still relevant, so we surface this information to the user in the form of a tooltip. When users hover over a stacked bar, the tooltip displays the total number of games represented, as well as the exact win and draw rates. We also put the raw number of games, along with its proportion of total games that the current graph represents on the bottom of the graph at all times. This way, we can still focus on the win rate but also inform the user if the moveset is common or not, allowing them to look at the graph with a more analytical eye and encourage them to use the tooltips. Justin implemented the animated stacked bar chart in about 3 hours. Based on peer feedback, we realized that the win/draw rates for the black pieces are also of interest. Thus, Justin added a drop-down selection embedded in the title of the bar chart, allowing the user to explore the win and draw rates from both player perspectives. We also realized that the Elo rating system was unintuitive for people unfamiliar with chess, so we added “Amateur” and “Expert” directional markings on the x-axis to inform users that higher ratings correspond to higher skill. These additional features took Justin about 1 hour.

The chessboard and graph components were made in parallel, which is why we designed them to only need a simple input/output. The PGN being a widely used notation in the chess board and being readily usable in the library and dataset made it the obvious choice. The chessboard outputs a PGN based off the user input, which then gets sent to a function that filters the dataset to every game with a PGN that starts with the said PGN, then calculates the most common move and everything needed for the graph, which it then creates via D3. This way integration was easy and we could style and increment on top of it with relative ease.

Given a sequence of opening moves, our visualization looks through the data to identify games matching that opening. Because the complete dataset of chess games from February 2021 has size 24.4GB, we had to reduce it for efficiency, preferably under 100MB to host it in the same Github repository. Justin spent 1 hour on a Python script to parse the first 1,000,000 games, keeping information on each player’s rating, the result, and the sequence of moves played. We used regular expressions to filter out irrelevant information, such as clock status and move evaluations. If we are able to continue this as a final project we would like to be able to work with more of the data. For the sake of efficiency, we only used a small fraction, but if it were to become a more significant project we would want to work with more. We would also like to add more features like allowing the user to input their own rating to see the most common move for their rating, making the experience more personal, and providing more metadata about the moves that they are making, such as explaining if it is part of a specific opening.