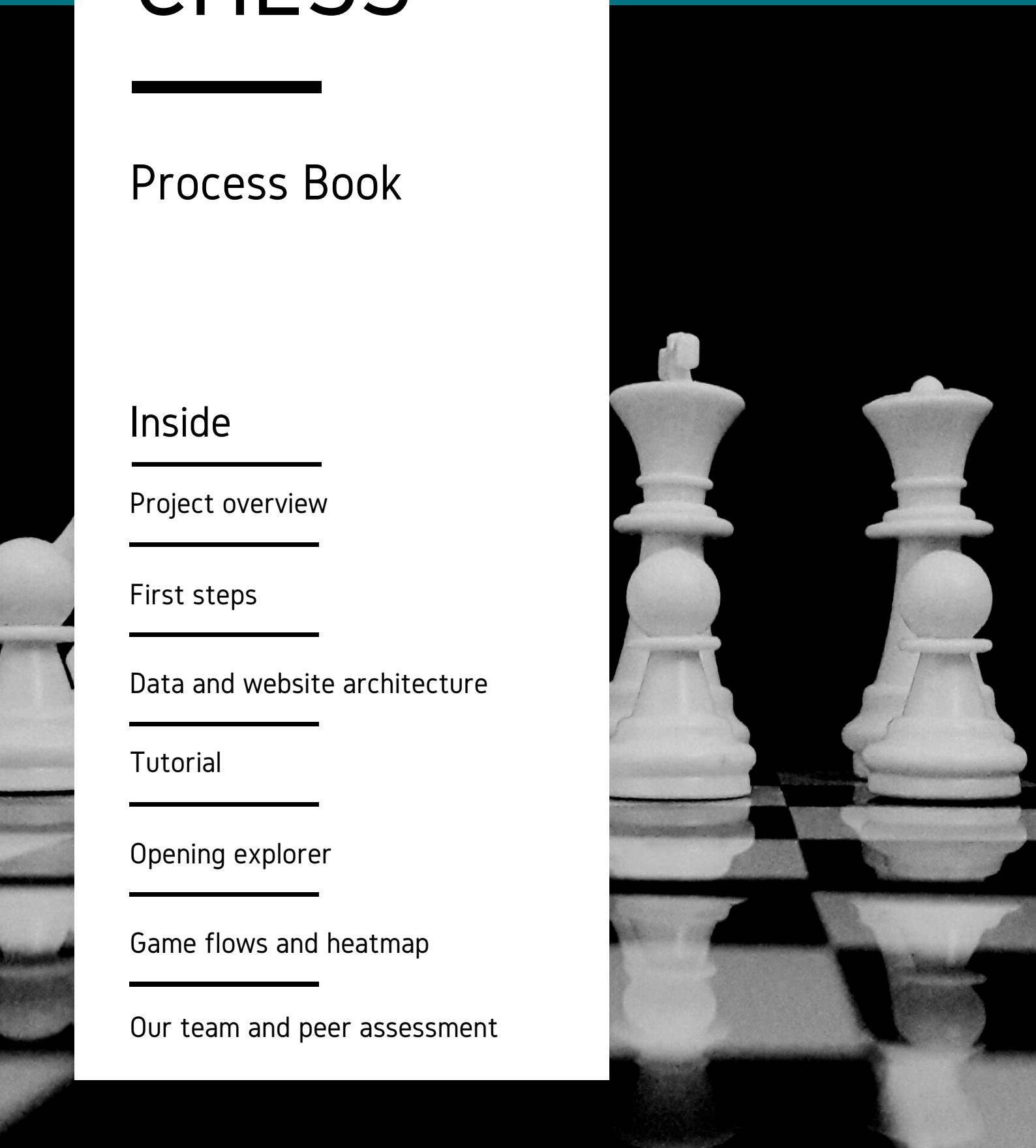
CLEAR CHESS

Process Book



Inside

OVERVIEW

Clear Chess is an interactive website uncovering the mysteries of chess for all types of players. This project was carried on as part of the course COM-480: 'Data Visualization'

The most important specifications that we took to heart throughout the conception of this project were:

- The accessibility; this website is destined for all audiences: beginner players as well as more advanced players have something to learn from this work
- Interactivity and intuition: our designs and visualizations follow the rules of simplicity and minimalism. Very easy to use and interpret, the different sections of our website offer neat visualization, aggregating a large and noisy dataset.

The website contains three principal visualizations (by order of complexity):
- Tutorial
- Opening Explorer
- Game flows and end position heatmap

This process book will detail why and how we carried out these different components.

FIRST STEPS

THE WORLD OF CHESS

We chose to explore the world of chess as it is a game that the members of our team like to play and watch. Moreover, it is a discipline that is inherently visual and has complex dynamics and tactics that can be explored and visualized.

OUR CONCEPT AND CORE IDEAS

We found a very rich dataset of 20,000 games collected from the popular chess website lichess.com. The strength of this dataset resides in the great number of games, the diversity of the players' levels, and the completeness of the information provided for each game. We wanted to exploit both **individual** games and tactics found in the dataset as well as let the user be able to grasp common trends and patterns from all **20,000 games**. In our reflections, we always kept in mind that this website should be interesting for all different types of players - from **beginners to experts.** The first part of the development process was therefore to brainstorm different visualization ideas, keeping in mind these two specifications. We gave ourselves no technical limits and didn't concern ourselves with feasibility to avoid hampering our creativity.

FIRST IDEA

Looking at different popular chess websites (namely chess.com and lichess.com), we decided to begin our website with a **tutorial** (page 4).
The tutorial is aimed at novice players to help them understand the rules of chess and let them see which moves each piece can make in order to better understand the following visualizations. Moreover, this module would serve as an introductory module for us to code and create the base visualization and core that we will use throughout the website.

SECOND IDEA

Our second main idea focused on openings.
The opening is a key component of a chess game, and we wanted to incorporate both the scope of an individual game (displaying and explaining one opening at a time) and the collection of games in our dataset (by showing statistical indications of each opening). These types of **opening explorers** are quite popular in chess websites and useful to learn new tactics, but we wanted to expand the core concept by adding insights on how these openings are played, by what kind of players and with what outcome.

THIRD AND FOURTH IDEAS

These are the most **ambitious.** We hadn't found this kind of visualizations anywhere else, so we had to conceive and plan the idea from scratch. We really wanted to use the vast collection of games we have at its best. The first module is a **game flow visualization**, allowing the user to visualize, for a selected piece, all the games unfolding in parallel. Flows move across the chessboard at each time step, as this piece is moved in the different games, highlighting certain common moves and patterns. The second module is an end position heatmap where, for each piece again, the player can visualize where it ends most frequently when the game ends. For both modules, we want the user to be able to select which kind of game they want to visualize, by selecting the ELO of the players (their level) and the outcome of the game (white wins, black wins or draw). We want these visualizations to be both impressive, aesthetic, and insightful on the unfolding of chess games.

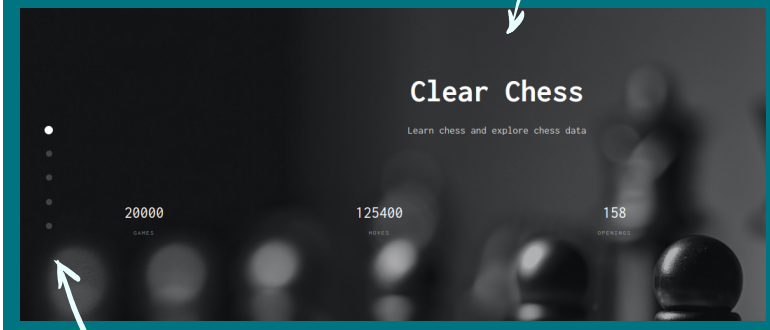
DATA AND WEBSITE ARCHITECTURE

WEBSITE CORE

The three visualizations having at their core a chessboard, we quickly came to the conclusion of having a website composed of three panels that we could scroll through: one for the tutorial, one for the opening explorer, and one for the game flows and heatmap combined. Each panel has a similar layout: a chessboard in the center with text and a button on the side, but each will have a dedicated color scheme.

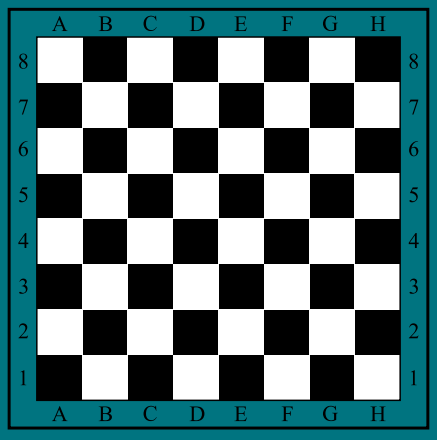
We use a free HTML open source framework as a skeleton that we tweak and adapt along the way.

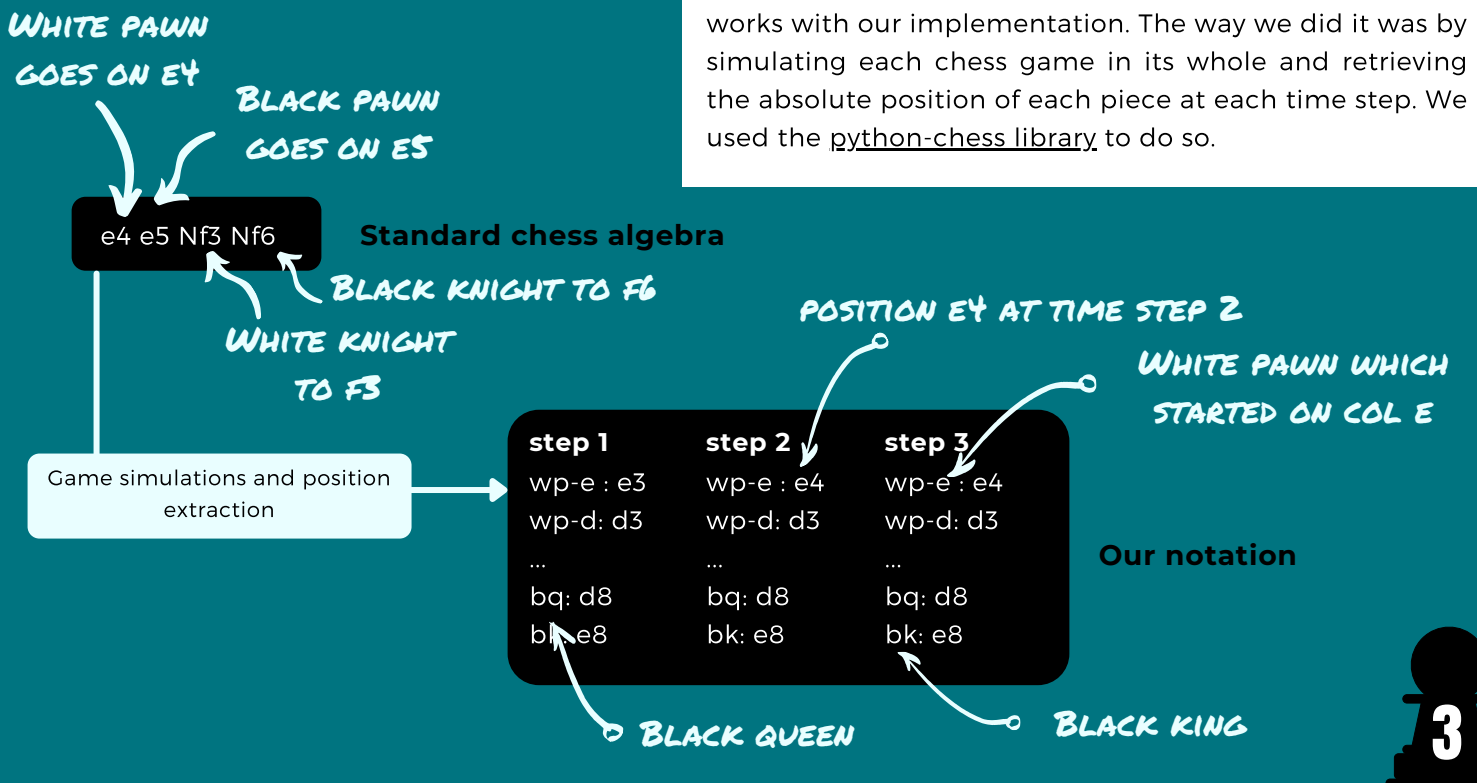
Using such a framework was a good push to start the project, but it later made the project more complex when we had to adapt it to our visualizations.



BASIC CHESS BOARD AND PIECE MOVES

The first step was then to create a chessboard, position the 32 pieces on it and find a way to encode their movements. We chose to give a unique id to all pieces and move them according to their absolute position on the board (more details on the JavaScript implementation next page). This poses our first big challenge. The standard chess algebra notation (which is the one used in our dataset) gives relative indications on the movement of pieces at each step. Which color and which exact piece moves is implicit in the notation and depends on the current configuration of the board. We therefore needed to translate this notation into one that works with our implementation. The way we did it was by simulating each chess game in its whole and retrieving the absolute position of each piece at each time step. We used the python-chess library to do so.



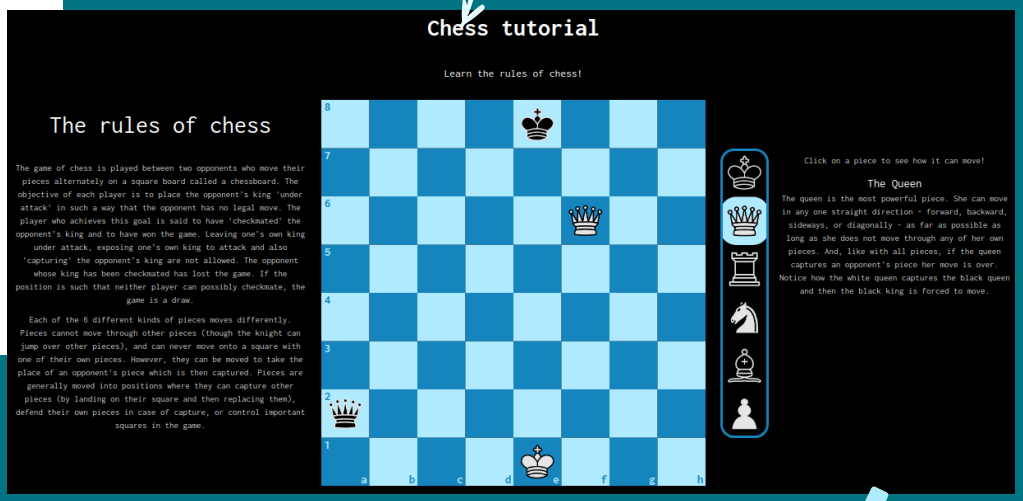


TUTORIAL

WHAT IT IS

The tutorial is the first interactive element the user is presented to when entering the website.
It is designed for beginner players and is not intended to teach users how to play chess, but rather to give them the basic knowledge necessary to enjoy the rest of the website.

FINAL VERSION OF TUTORIAL PANEL



Step 1: Create the board

Step 2: Write the tutorial

Step 3: Combine

We implement our own chessboard using javascript and d3. In the class Chessboard, the tiles are represented as **svg rectangles** and row labels as text elements, drawn once in the constructor. The chess pieces are represented as **images,** whose sprites are taken from the web and are drawn in the method drawPieces, according to their absolute position on the board (as discussed above). We knew that the Chessboard would be used in all three of our pannels, so we had to immediatly make implementation decisions as on what should be in the class Chessboard and what should not. We decided that **all the functionalities which drew directly on the chessboard should be kept inside the class and the others outside.**

Our first thought was to implement an interactive chessboard that would let users experiment with each piece, which moves they could play and how to deliver a checkmate. We quickly realized that doing so would require creating a small chess engine that would enforce the rules and identify patterns. As the scope of our project was to visualize games and not teach how to play chess (and there already exists a lot of websites doing this), we decided to make a much simpler tutorial. We wrote a brief description of the game, and for each piece, a description of its legal move as well as a little live simulation to illustrate the moves.

To make the interface simple and easy to use, we let the user select the piece they want to learn about in a menu on the side of the chessboard. We use a blue color scheme for this section and incorporated little touches of blue in the different elements and button for visual harmony.

OPENING EXPLORER

WHAT IT IS

In chess, the first moves are crucial for creating an advantage in the game: they are called **openings**. Openings have been named and studied for centuries and constitute a big part of theoretical learning for chess players. Our visualization proposes to explore the most popular openings in our dataset. The user can see the unfolding of the theoretical opening as well as get insights on how effective an opening is and what kind of players use it.

FINAL VERSION OF OPENING PANEL

Step 1: Conceptualize and sketch

We asked ourselves two main questions for this part.

How do we make the module interesting to all players, independent of their level?

We include a diversity of elements. For beginners, we have an explanation of what is an opening, and why it's important. For more advanced players, we have the standard chess algebra of the opening for them to write, and statistics on the opening to help with tactical choices. For those who just want to discover new openings, we have a 'random opening' option.

How do we display the opening while giving the user control?

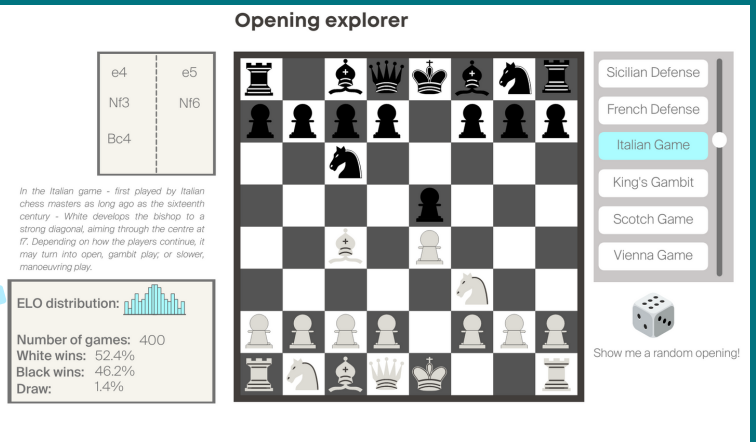
We allow the user to either 'play' the opening and see a simulation on the board, or to control the opening step by step with the two buttons 'back' and 'next'.

Step 2: Basic implementation

For each game in our dataset, we extracted the opening name, its moves, the positions of the pieces at each timestep, the game outcome, and the player level. We aggregated the results in a JSON file to use directly on the website. In our first version, the player could only select an opening and play it all in one go (using the d3 interval function to update the state of the board each second).

Step 3: Improve UI and design

We chose a pink/red color scheme and adapted the buttons' style accordingly. We changed the spacing between the different elements and put all the static elements (the text) on the right and the dynamic elements on the right.



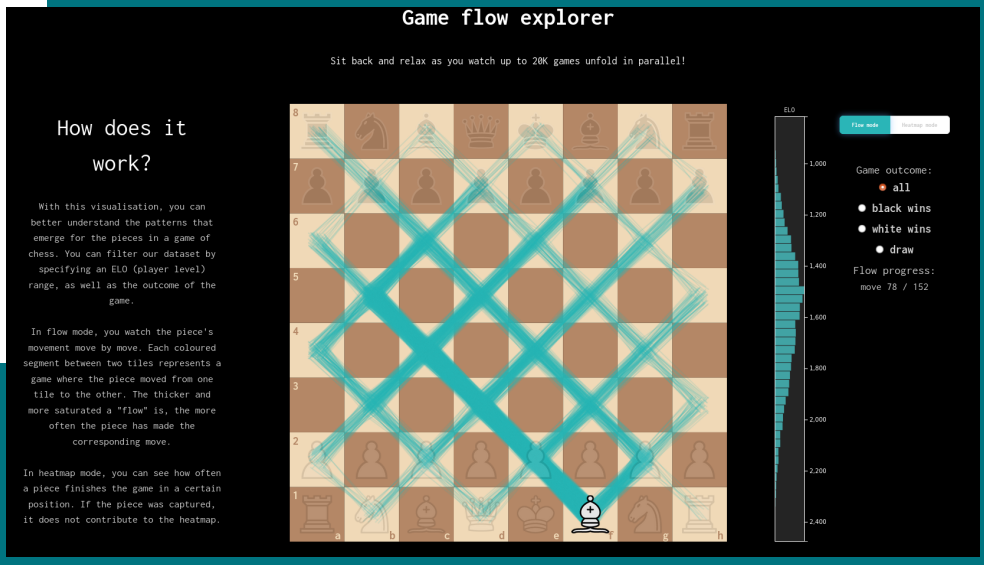
FIRST SKETCH OF OPENING EXPLORER

GAME FLOWS

WHAT IT IS

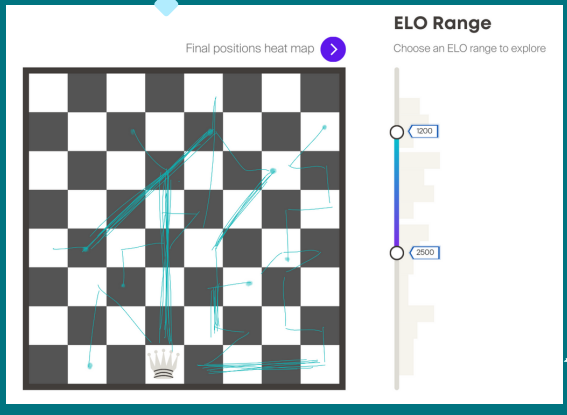
This innovative module brings together many games of our dataset in a single visualization. It shows the trajectories of a selected piece in multiple games, step by step. We use 'flows' of color to visualize this, in which common trajectories appear bolder and wider than the others. Also, this visualization is not static, to show that pieces behave differently at different stages of the game.

FINAL VERSION OF GAME FLOW



Step 1: Conceptualize and sketch

We have the general idea, but we need to decide what degrees of freedom the user has on the visualization. There is a trade-off between powerful visualization and a confusing UI. We chose to allow three degrees of freedom: which piece, what ELO range and what game outcome can be controlled. At any moment when the user changes one of these parameters, the simulation starts over.



FIRST SKETCH WHEN WE WERE ONLY THINKING OF 2 DEGREES OF FREEDOM

Step 2: Basic implementation

For the visualization to be responsive and efficient, we aggregate the data based on ELO and game outcome before drawing the flows. We also index the data by time step, to allow the iterations to be as fast as possible.

While simulating each step of the game, we have way more flows (state changes) at the beginning of the game than at the end. This is because as time goes on, more and more games have already ended. This gave the visual effect of a very fast beginning and then a slow ending (with very few changes). To palliate this, we increase the speed of the simulation as time goes on, following a transformed sigmoid function.

Finally, to create the visual effect of proportional thickness of the flow depending on the frequency of a certain move, we introduced a small random jitter that gets added to the start and end position of every line of the flow.

Step 3: Improve UI and design

Once the basic elements were coded, we chose a neutral beige color to contrast with the sharpness of the blue flows. We added an ELO histogram to the ELO range selector, and some text on the left to give balance to the screen and explain how to use the module.

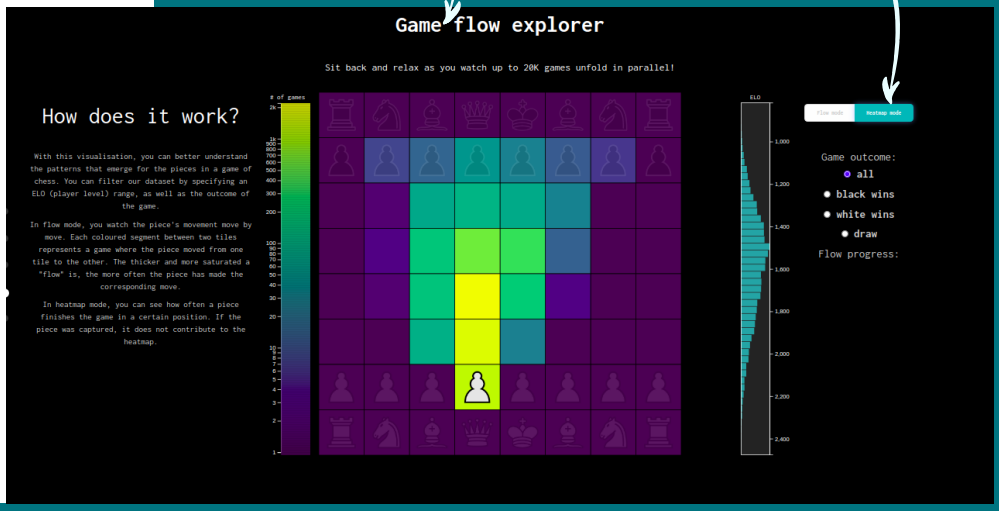
HEAT MAP

WHAT IT IS

This creative module allows us to visualize the heat map of the end positions for each piece on the board. This allows us to see how much each piece moves during the game and the parts of the board that it can reach. The ELO bar allows us to visualize the heat map at different levels.
It is interesting to see that players with low rank tend to underuse certain pieces while highly ranked player tend to use pieces to their full potential.

Final version of heatmap

Toggle button to switch modes



Step 1: Conceptualize and sketch

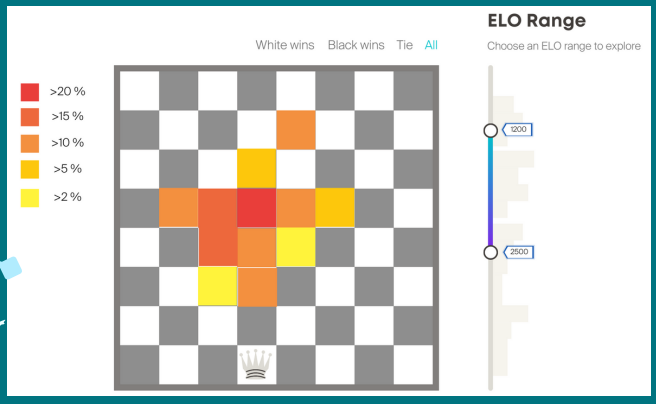
This visualization was first thought to be the ending of the game flow visualization. We pictured that at the end of a flow simulation, the board would turn into a heatmap to reveal the most frequent positions.
We later decided to make it a visualization of its own as we don't necessarily need to see the flows before seeing the heatmap. The main challenge was how to show the data on a board that already contains squares, of different colours? We thought about having circles or smaller squares inside the tiles, but the alternating colours of the tiles gave a weird visual effect and some occasional optical illusions. We finally came to the decision of entirely covering the tile with a coloured square.

Step 2: Basic implementation

We extracted the ending positions of the dataset, aggregating them by ELO and game outcome. We then implemented the heatmap, using the Viridis color scheme which is color blind friendly and has a good distance measure, adding a colorbar for easy interpretation. We use a log scale for the colours so that it is easier to notice the differences for less frequent positions (and as there are 64 possible positions, there are a lot of low frequency squares).

Step 3: Combine with data flows

Because the heatmap and the game flows have the same degrees of freedom, we implemented a toggle button that allows to go from one mode to the other, while keeping the already selected parameters. This makes the module very intuitive and allows switching from one mode to the other easily.



Sketch of the end heatmap

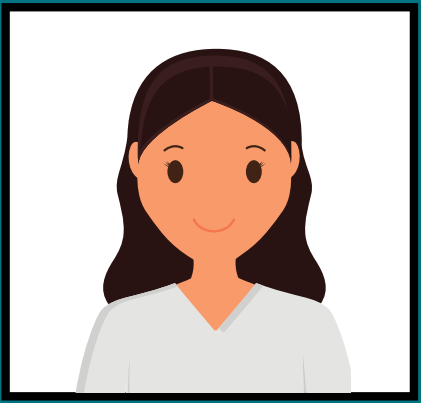
OUR TEAM

Like in all good teams, each member had a domain of preference and specialized in it:

LUÃ STREIT

d3 Wizzard

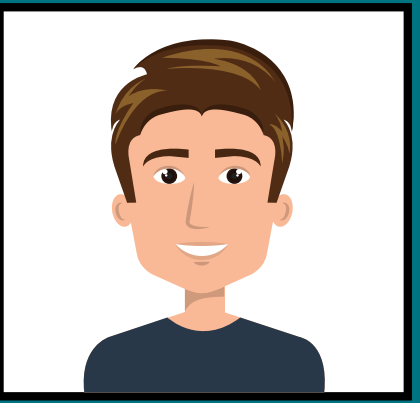
- Chess amateur (low ELO)
- Researched viz ideas.
- Pre-processing of the data.
- Implemented the chessboard, the piece's animations, the tutorial, the openings, the flow, and the heatmap.
- Allowed each visualization to be interactive through buttons and other components.
- Worked on the UI.



MARIE REIGNIER TAYAR

The designer

- Chess amateur (ok ELO)
- Our design expert!
- Pre-processing of the data.
- Translation from chess algebra to our notation
- Sketched the visualizations.
- Implementation and UI decisions
- Wrote this process book
- Wrote the screencast script



DAMIEN GENGLER

UI expert

- Occasional Chess Player
- Tried out several website designs.
- Worked on the UI.
- Developed the website's browsing experience
- Integrated the visualizations in the website.
- Wrote the screencast script

CONCLUSION

We are happy with the way our visualizations turned out, especially the last two ones.
We feel that we have gained some interesting new insights and hope to have become better chess players.
Sadly, we had to abandon many of our ideas, and underestimated the effort needed to make the visualizations intuitive.
If we had to do this project again, we would improve our workflow, as many of our tasks were interdependent, and sometimes we had to do repeated work.