SHEGO (Simple HEatmap GO)

[Motivation, audience, goals, research question, hypothesis]

1 The Project

1.1 The Motivation and Audience

Go is a skill-intensive game that requires cognitive skills similar to Chess, the difference being that pieces can be placed almost anywhere on the board, resulting in a much larger "move pool" compared to Chess. Because of this, many people become intimidated at the concept of learning the game and are easily overwhelmed with the game's possibilities. People who do overcome that and continue learning, however, are met with a steep learning curve that requires a lot of studying from semi-limited resources. Those people are our audience. We would like to develop a tool where these Go students can access these resources easier and learn faster.

Our primary motivation was to provide a tool for learning Go strategy. One of the common methods of learning higher level Go strategy is to study other players games, much like chess. Our motivation comes from desire to be able to study these games efficiently alongside wanting to see if comparing multiple games together could teach a Go student just as well as reading single games.

1.2 Research Question and Hypothesis

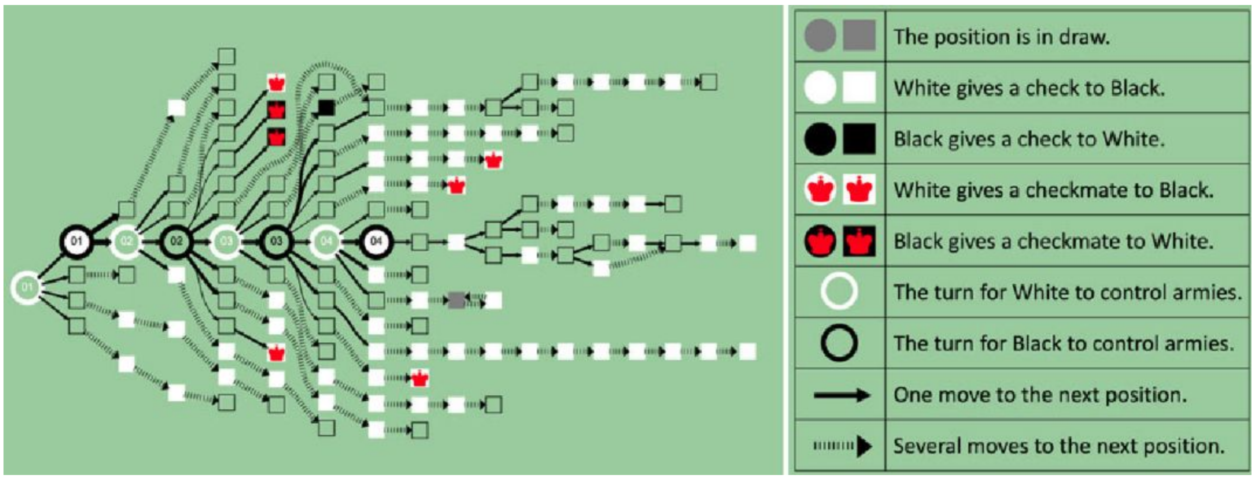
So our question is: can we develop a visualization tool that will help Go students learn and understand Go strategies better? We hypothesize that this tool, if effectively developed, will aid in the learning of the techniques in Go by letting the player simulate their own game while changing the board's visuals to communicate how the game is progressing. Our target audience would be people who have begun to learn higher-than-novice level strategies in the game, these people have a novice level grasp of the game and are looking for learning higher skilled techniques to further build their skills as Go players.

2 Related Work

There are Go learning tools in the internet, a popular example is the website, PlayGo. PlayGo is a site that acts like an interactive E-Book that teaches the fundamentals of Go. In the interactive portions of this site, there are sections of "games" that the reader can then try to deduce what the smartest next move is, if the reader is wrong, the game will reset, if right, the game will end (Mori, playgo.to/iwtg/en/). The reader can also just skip these sections if they so please. We would like to emulate the simulation aspect of this site, but not the testing side.

Another tool that we drew inspiration from is from a paper titled "Chess Evolution Visualization". The authors created a tool to further understand the complexities of Chess (the only other board game considered to be "high skill only", alongside Go). Their approach revolved around having their tool predict and highlight future moves and strategies of a game of

Chess. They never actually properly visualized the tool on a board, but created a large visual tree to communicate the predictions. Another paper, called “Visualizing Chess Games”, had a similar approach but actually visualized it on a Chess board. The authors created a tool that selected a certain piece and projected colors to communicate which move is “safe” for that piece. It also had a simple prediction system, less complex than the previous papers, but there nonetheless.



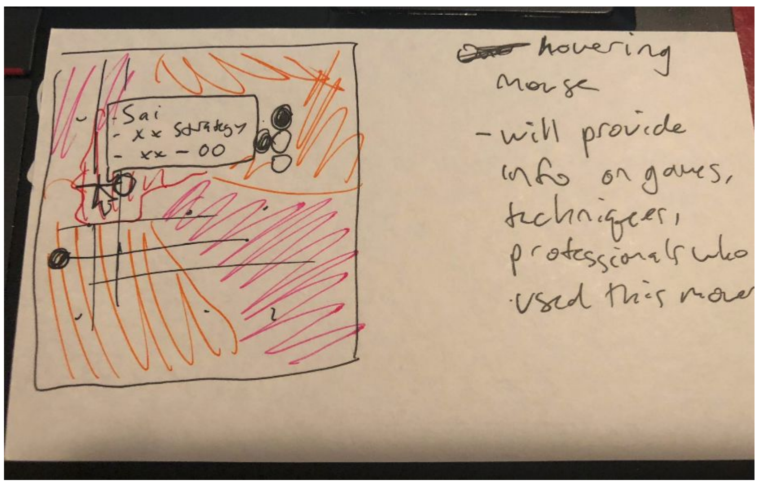
This image is a visualization of Chess Evolution, our tree data structure was inspired by this (Lu, “Chess Evolution Visualization”).

This is from Visualizing Chess Games, our simulation was inspired by this (Lewis, “Visualizing Chess Games”).

We would also like to communicate future moves, but we can’t develop a tool that would actually predict good strategies. Go is largely unsolved, even with the existence of AlphaGo, simply because Go has such a large move pool. So instead of making an AI tool to predict moves, we will access a large database of recorded games to show the user commonly used moves. This is a rather niche project, but our shared interest in Go made working on this rather fun. Go is one of the most complicated games out there, and making a visualization of thousands of games, we think, can possibly lead to discovering more about the game to the average player.

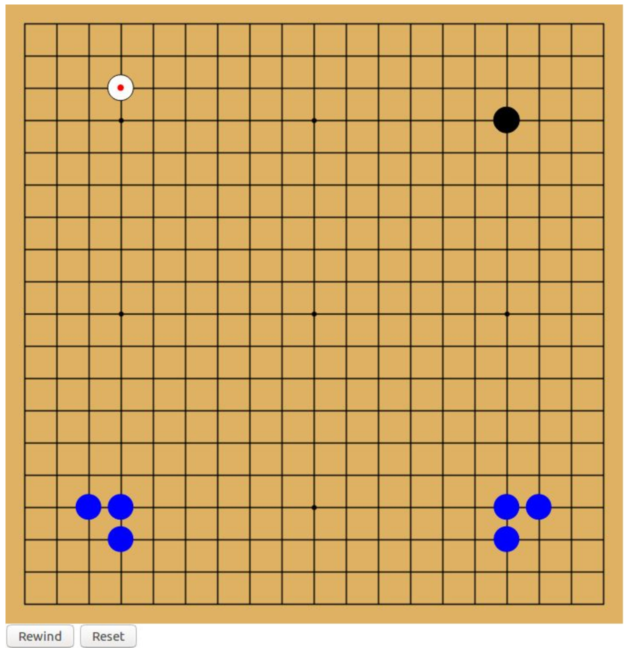
3 Evolution of the Design

Our design was simple from beginning to end. Our initial sketches were a bit messy but wanted to show one point: where the next stone is placed and how often. The visualization revolved around that principle. As we were sketching out the design, we were collecting data from an online source (Li, go4go.net). We collected around 75000 games in total, but we were only able to visualize 30000 due to machine constraints.



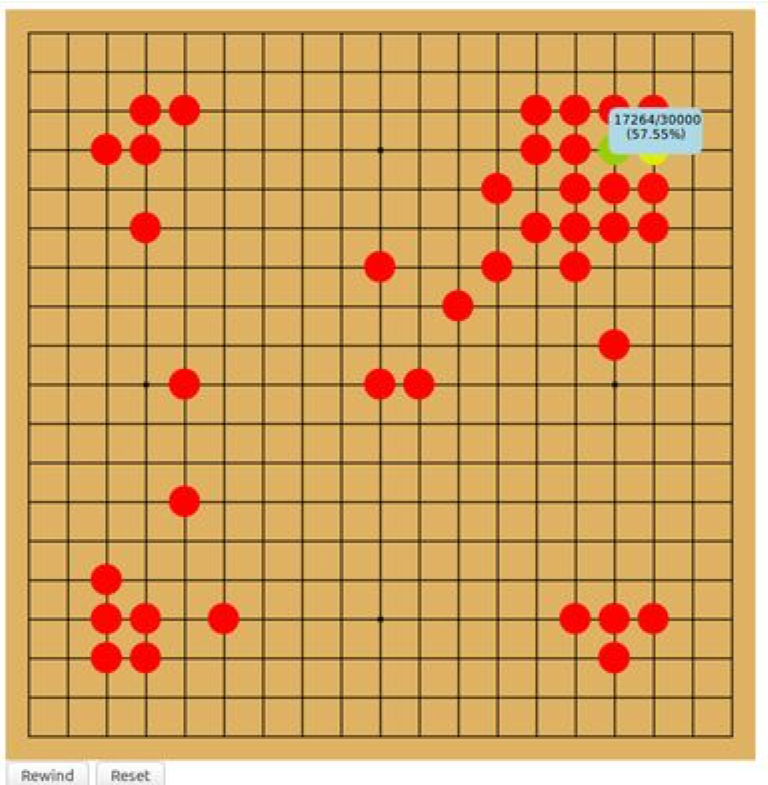
Our initial sketch with hover over for the future stone placement.

From there, we started creating the front end of our simulation. We did not decide on a color palette right away, so our initial “heat map” was just a solid blue, the stones were just the original black and white.



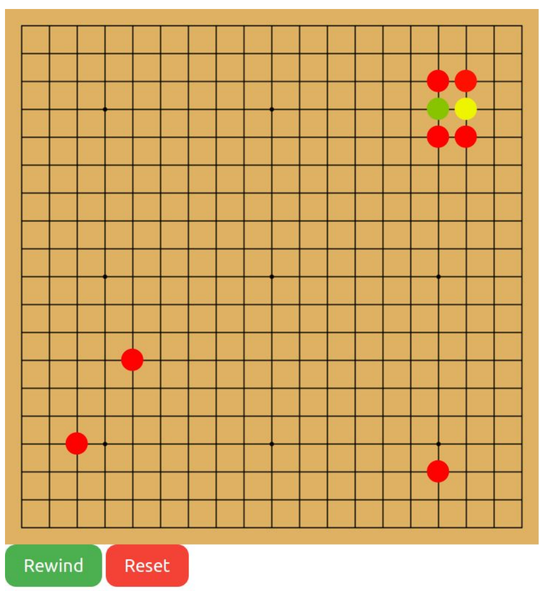
Initial heat map implementation, recently placed stone has a red dot.

We also have working rewind and reset buttons for the user to clear the board or step back when needed. We then implemented the colors, we decided to interpolate from green - yellow - red, green symbolizing most common moves and red being least. Hover over for each stone was also added to give the user more insight on how games have gone. We tweaked the values for scaling, but a lot of moves ended up being bottlenecks to single games, so our heatmaps ended up like this:



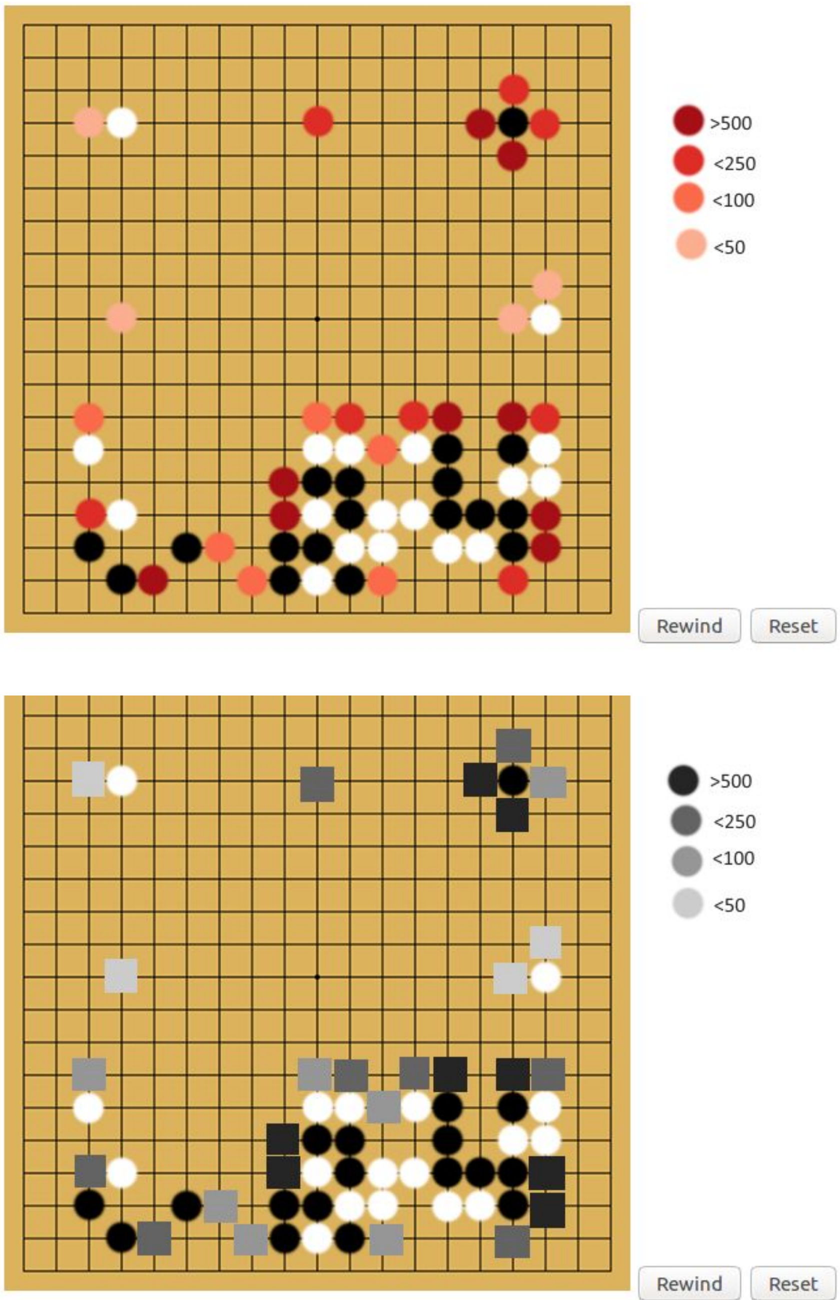
Heat map coloring, hover over implementation included. This visualized 30000 games.

Below is the current version of the simulation, pretty buttons included.



4 Feedback and Changes

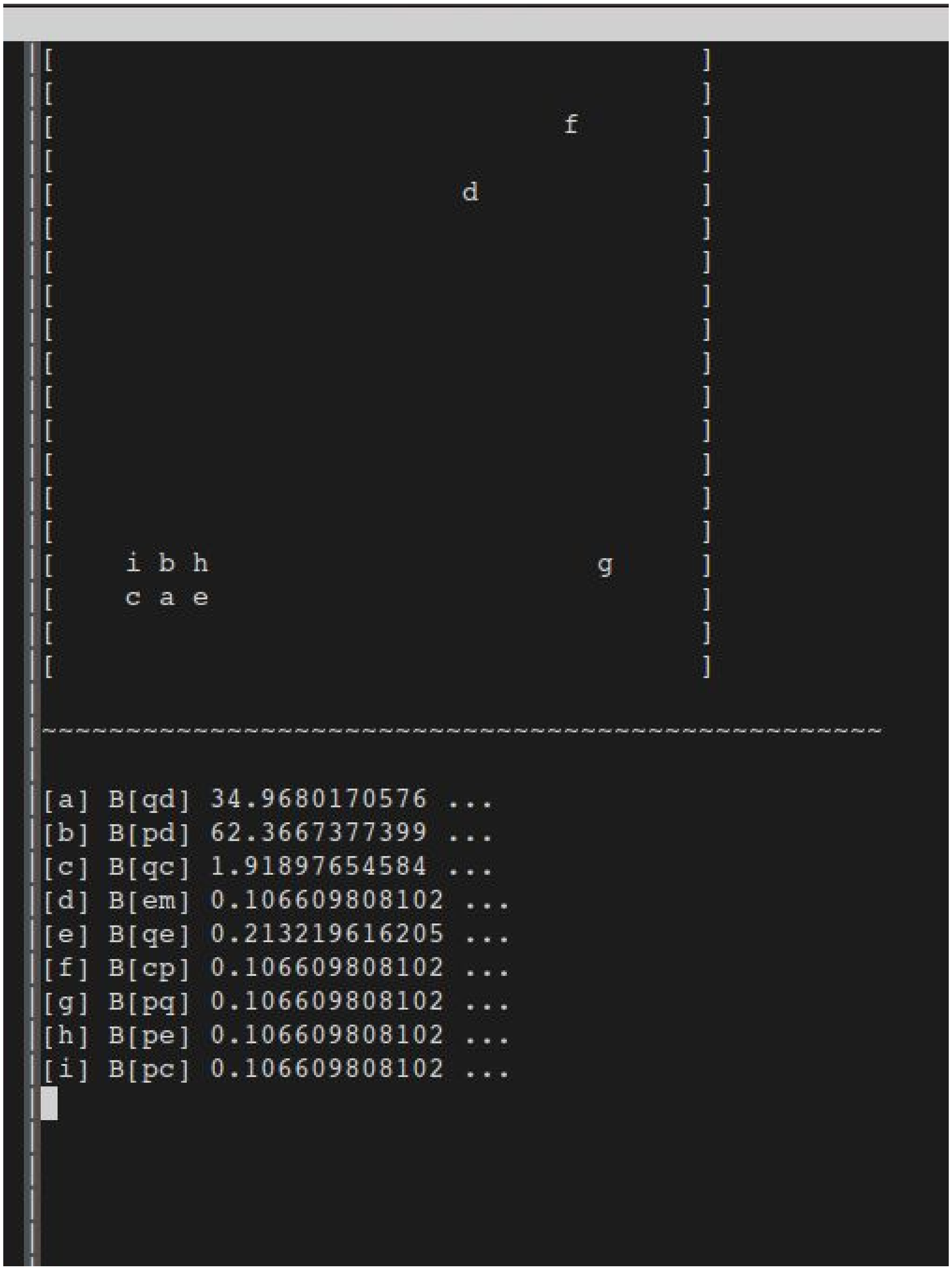
We got feedback on two aspects of our design: heat map representation and user interaction. For the heat map representation, we showed off a number of different mockups of our design using, shape, color, and opacity to encode how often a move was used. Here are some of our initial markups:



Two heatmap considerations, red was the most popular.

We found that people preferred the circles to the squares. We also found that opacity was harder to interpret than types of colors. One thing we noticed was that with the color, we were changing the value, which would cause parts of the heatmap to fade to white. We were concerned that this would cause users to confuse potential moves with stones already placed onto the board. When we showed this to users we found that they also shared this concern. So we instead used a color scale which interpolated from red to yellow to green to show the heat map. This way we avoided the problem of the heat map fading to white and still convey percentage information with color

Next we showed off a text based mockup of our tool. We showed the users a text based version of the board and presented them with a number of possible moves to choose from, allowing the users to play through our conglomerated game tree. Users generally liked this style of interaction but it left some things to be desired. One was that having a GUI interface and a proper heat map would be much nicer to use, which we planned on implementing anyway, but it was good confirmation that we were going in the right direction. Additionally, users wanted more data because the tree the users were exploring was rather sparse. So we went in and tried to cram as much data into the tree as we could without slowing our tool down to a crawl. Some users suggested an autocomplete feature which would autocomplete a game if we reached the case where they were only exploring a single game. This was a feature we would love to add but we didn’t have the time.



Our text based prototype

5 Implementation and Features

Our core features are the placing of stones and the heat map itself. The board will display all the potential moves to explore along with a color indicating how often those moves were played. The user can then select one of those potential moves and see what moves were commonly played from there. Additionally, the users are able to rewind back one move and to reset the board to the beginning.

For our implementation, we used the sgflib Python library (source) to parse all the SGF game files. From there, we took that data and built it into a tree of possible moves which could be traversed based on which move was selected. The board was implemented using d3 to place the stones and draw the heat map.

6 Conclusion

Go is a very complicated game of skill, but that skill does not mean uniqueness in play style. As we were developing this visualization, we have discovered that many games start and continue similarly for a few rounds, before deviating and becoming their own unique game. Bottlenecks are often, but games that branch from long bottlenecks do happen, studying the board state at these parts is very important, as it shows definite strategy that all players should know of and play. In the future, we would like to show this tool to more people who play Go and see what kind of strategies they would be able to see out of viewing so many games.

Work Distribution

Lorelei did all of the data parsing and formatting, as well as front end bug fixing.

Annie did the data collection and front end simulation.

Bibliography

Lewis, Scott & A Voida, Stephen & Martin, Chris. (2018). Visualizing Chess Games.

Lu, Wei-Li & Wang, Yu-Shuen & Lin, Wen-Chieh. (2007). Chess Evolution Visualization

Mori, Hiroki. (1997). http://playgo.to/iwtg/en/

Li, Mace. (1999). go4go.net/go