

ChessBrancher

My History with Chess

While I was growing up, my father taught me how to play chess. My brother, my father, and I would regularly hold little tournaments and the winner would be called the household champion. It sounds kind of silly reflecting on it, but I really enjoyed it and it holds a special place in my heart.

Beyond that, I don't have much experience playing chess. I am an average player at best. But, I love discrete mathematics. In particular, I have a passion for graphs and trees. For the past couples of years, I've been exploring the possibility of expressing the chess move history in the form of a graph.

My Friend Tamal

My friend Tamal has become quite successful in AI research on human vs computer chess play. He analyzes thousands of games. Ultimately, he rates the individual moves and finds the positions where a bad move causes a turning point in the game.

Tamal once suggested to me that analyzing a particular board position and having the freedom to test out many different gameplay branches is quite valuable. The many programs that he uses do a great job, but there is something valuable about having the human ability to traverse the gameplay tree by hand. Visualizing a hand selected traversal of the gameplay tree becomes a necessity.

My Project

To allow users to create their own multi-path traversal of the gameplay tree, I created a little web app called [ChessBrancher](#). The app allows users to enter a chessboard position in the form of a [FEN](#) code. Then, users can try out as many possible moves as they please, building a tree of possible gameplays.

The app utilizes [ChessJS](#) and [ChessboardJS](#). ChessJS is great for determining the possible valid moves while ChessboardJS can elegantly produce an interactive chessboard. Both of these libraries have wonderful development communities that I strongly encourage developers to get involved with.

Get Involved

ChessBrancher has only just begun and I am asking you to get involved and contribute your development skills. You can find our current code [here](#). We already have a list of at least 8 suggestions. Please join in and tackle one from the list. Just make a pull request to get started.