Motivation

I have used extensively the best chess web sites out there including chess.com, lichess and chessable as well as the desktop software from ChessBase. While I am tremendously enjoying all of those tools, to study chess exactly the way what I wanted, I had to come up with a process that combined their capabilities with extra manual steps.

Becoming tired of that manual process, I have decided to write a desktop application that would automate it. And that’s how the idea of Chess Forge was born. And I’ll add upfront that this is a completely and unconditionally free Windows application for anyone to download and use as they please. It is also Open Source so developers can contribute if they wish to. A note to those developers, in its current incarnation this is a Windows desktop application written in C# using the WPF framework. Porting it to Web at some point in the future may be an option.

So, what is this missing scenario that I was talking about. Well, I had trouble learning openings, mainly due to my poor memory but also because of how easy it was for me to drop into this mechanical learning mode that where you really lose interest in the thing and as a result don’t absorb the knowledge.

I have tried chessable.com which is a great site with great content and uses their patented “spaced repetition” methodology. Unfortunately, that method just does not work for me. I have found the mechanical nature of force repetition, when I actually did not want to keep repeating the moves, very annoying. So, I guess, it works for some, does not for others.

The method that I have found to work perfectly for me was to build, or obtain, an opening tree like you can build as a study on lichess, or as a database in ChessBase and then test yourself against it.

One aspect of the test was to check if I remembered what I learned, and the other was to verify if I understood it. The first test simply checks if your move was in your opening book indeed while the other I tested by playing from a certain position in the book against the computer.

I was copying critical positions from my opening book into a computer game on lichess and played it from there. I was excited to see that lichess appears to randomize its engine’s responses and, therefore, could more comprehensively challenge me.

I got finally convinced of the value of this approach when studying French Rubinstein for Black. Two memorable things happen.

First, in the main sub-variation, considered and evaluated by the engines as equal, I got very quickly checkmated by the computer. A few short minutes of analysis revealed that I was one move too late placing my rook on the open d-file allowing White to mount an effective attack. Lesson learned!

In the second instance, the engine played a non-book move Ng5, subsequently sacrificing the knight on f7. I made 4 or 5 fairly accurate moves but then succumbed to the attack. Again, analysis with the engines showed that the sacrifice is not correct but requires precise play from Black. Well, that game, along with the analysis, made it into my Opening book.

And I got highly motivated to develop a tool that supports this method of learning with all the helpful bells and whistles.