The Chess Tutor

John Ktejik

Georgia Institute of Technology

Kenosha, USA

Jktejik3@gatech.edu

ABSTRACT

This paper introduces an Intelligent Tutoring System, The Chess Tutor. The Chess Tutor has specific features that separate it from other chess teaching programs: plain English advice, advice when playing an entire game of chess, feedback and advice in real time, and conceptual advice designed to make the user think. The inadequacies of existing automated chess teaching methods are discussed, as well as summaries of existing research on teaching or learning chess. The Chess Tutor attempts to follow best teaching practices by modelling itself after traditional classroom Intelligent Tutoring System Interfaces, and incorporates enjoyable and motivating techniques such as avatars with personalities, and positive, empathetic statements. The Chess Tutor uniquely contributes to the field by showing that there can exist a standards-based format for teaching chess which has all the essential features of a human tutor.

AUTHOR KEYWORDS

Educational technology; chess; Chess tutoring; intelligent tutoring systems.

ACM CLASSIFICATION KEYWORDS

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

If learning chess were that easy, everybody would be good.
-Dan Heisman

This paper presents a new way to teach and learn chess: www.thechesstutor.com.

Figure 1 - The Chess Tutor

The interface (figure 1) is divided into two parts: main, and hints. The main interface is the chess board. Here a user can play a game of chess against the computer like normal. On the left is the hints panel. Whenever a user would like a hint on what to do next, they can press the hint button. Unlike traditional chess engine hints, where typically an arrow appears showing the best moves, these hints do not suggest a specific move. Instead the hints refer to a general concept. The first few hints are as general as possible. If the user does not understand them, they may click on the hint button again, and a more detailed hint appears.

ESSENTIAL FEATURES

Like its name implies, The Chess Tutor is meant to replicate everything a human tutor would do. This means The Chess Tutor has specific features that separate it from other chess teaching programs.

1. It gives feedback in plain English, not in code, numbers, or chess notation.

2. It allows the user to play an entire game of chess, as opposed to contrived scenarios.

3. It gives feedback and advice in real time. Analysis will happen automatically while the user is playing. It is not a traditional game analysis that happens after the game is over.

4. The advice is designed to make the user think. It will be high-level, abstract and conceptual. It will never suggest explicit moves.

Plain English

Although the requirement for plain English may seem obvious, historically chess analysis is done by writing out the game in chess notation, a compact form where merely the piece and the board square is listed. Modern chess engines give all their output in chess notation. (figure 2)

Figure 2 - A typical chess engine analysis in algebraic chess notation

While it is possible to learn from this chess code by replaying games and figuring out what one did wrong, this process is slow and still relies on extensive chess knowledge to realize what you did wrong after the fact. It is not appropriate for beginners.

Modern chess programs have realized the need for plain English and have supplemented their analysis with simple statements like 'this is a good move' or 'this was a blunder'. Chess.com, one of the most popular websites for playing chess, is a good example of this. (figure 3). However, the feedback is limited to just saying if a move is good or bad – there is no understanding or explanation of why a move is better.

Figure 3 - the chess.com analysis tool

Another popular commercial product that advertises tutoring in plain English is the German-based program Fritz. Cute icon notwithstanding, the feedback is merely a single sentence saying which side is winning.

Figure 4 - Fritz's "plain English"

The Chess Tutor does not use any algebraic notation anywhere. All advice and feedback is in complete English sentences.

A game of chess

If one were working with a chess tutor, or even participating in a class or chess club, playing a complete game of chess is the fundamental activity. A human tutor would work with you as you played, analyze the moves on the board, and suggest advice, regardless of the state of the game. Yet most online chess lessons force the player to start from a pre-determined position and play from there (figure 5).

Another popular chess site with an extensive library for learning chess is lichess.org. It has hundreds of exercises to help learn chess and chess concepts. Unfortunately none of these lessons allow the user to play a game of chess.

Figure 5 - a typical online chess lesson

Feedback and advice in real time

The idea of a computer chess tutor is not new. All the previous examples advertise themselves as chess tutors. They give advice, and it *is* in plain English, albeit simple good/bad statements. But one thing they do not do is give advice or feedback in real time. They expect you to make the moves, or even play an entire game, and *then* ask for help. This is backwards on how a human being would tutor. With a human tutor, they give advice first, and you make moves depending on their advice.

The Chess Tutor gives feedback before and after each move. Feedback before the move is called *advice*, and after the move is just *feedback*. Giving feedback on a move, before the computer takes their turn, is an important feature of The Chess Tutor as it models how a human tutor would operate.

Furthermore, anyone who has played a chess program has experienced the moment where you realize you are losing, and hit the 'undo' button and try again. It is a frustrating experience. The Chess Tutor attempts to eliminate this frustration by giving immediate feedback after a move is

played, and based on the feedback, the user can choose to undo the move or confirm their move, before the computer moves.

High level advice

The first three features – plain English, a complete game of chess, and real-time advice – are enough to uniquely separate The Chess Tutor from any other chess teaching program. However there is one more feature that is essential to The Chess Tutor, and that is the concept of high-level advice.

Many chess programs offer hints or advice. It may be in algebraic notation, or even in English. ChessMaster, one of the oldest and most enduring chess programs, has the ability to give detailed advice in plain English. However, in the first sentence, it tells you the exact move to perform. It is this sort of blind, low-level advice The Chess Tutor avoids.

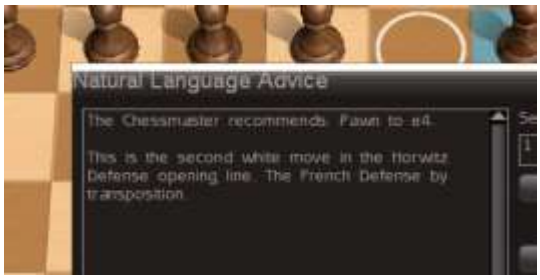

Figure 6 - ChessMaster advice

Another example is Chess Free, the #1 chess app on the google play store. It advertises itself as a chess tutor, but turning on tutoring mode does nothing except highlight the best move! (figure 6)

Figure 7 - The Chess Free app with 'tutoring'

These sort of hints are not conducive to learning. Anyone who has ever played a chess program with any sort of hint feature has found themselves abusing the hints. It's too easy to fall into the habit of playing the move without really understanding or learning the concepts behind the move. You *could* learn from these suggested moves through careful study, but often the motivations behind the move are elusive. Following the computers' advice blindly helps you win, but it does not help you learn.

To avoid this dependence on the hints, The Chess Tutor shows high level, abstract concepts, like controlling the center and getting your knights and bishops out first. Hints are designed to start from the most general 'you have a good move available' and progress to more specific 'why don't you go ahead and take the free piece'; but it will never tell users exactly what move to do.

INTELLIGENT TUTORING SYSTEMS

The Chess Tutor is modeled after a type of program known as an Intelligent tutoring system (ITS). ITS fall into the domains of Artificial Intelligence, Education, and Psychology. Artificial intelligence, because the program has to learn (adapt); Education, because the student has to learn; and Psychology, because of the questions ITS raise about how the mind works and how learning takes place. ITS and researchers make use of all three disciplines to explore and improve student's learning.

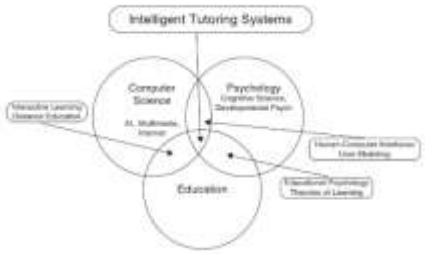

Figure 8 - The three domains of Intelligent Tutoring Systems

Intelligent tutoring systems have a few key points, the most important being immediate feedback, and adapting or changing its behavior in some way depending on the user [17]. These key features mesh well with the goals of The Chess Tutor and serve as an appropriate teaching model to use.

Modeling The Chess Tutor after an ITS has the main benefit that an ITS is a known, reliable way of teaching. ITS generally look very similar, with the screen divided into two parts, one for the model or problem they are solving, and the other half for inputting their answers. (figure 9). This greatly simplifies the design process and provides a general framework on which to build.

Figure 9 – the AutoTutor interface. It has two main parts – advice and an accompanying avatar on the left, and the display and interactions on the right.

There are no strict definitions of what defines a particular type of ITS, but ITS can be classified into some general types such as cognitive tutors, model-tracing tutors, and constraint-based tutors. The Chess Tutor models the *constraint-based* type of ITS. Simply put, a constraint-based system asks the user to solve a problem or explain a concept, and to do so, the user must input or demonstrate knowledge of the intermediate concepts, or *constraints*, before the problem is marked as solved. The order the inputs or concepts are mastered does not matter [16]. This separates constraint-based from other types of ITS, such as the model-tracing system, which requires the steps to be sequential, or ordered in some way. [17]

In practice, constraint-based advice means The Chess Tutor will give a general hint but not give any specific order or route to satisfy its advice. For example, The Chess Tutor gives the advice 'get your knights and bishops out' (figure 10). It does not specify how to do this, or in what order; it only asks that it be done. When the knights and bishops are successfully moved out, in any order, another message will appear congratulating the user on getting their knights and bishops out, and the knights and bishops advice will stop appearing.

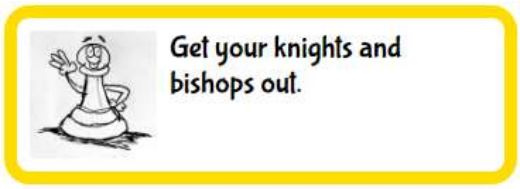

Figure 10 - The Chess Tutor advice

This constraint-based concept applies to almost all of The Chess Tutor's advice – castling, getting knights and bishops out, and controlling the center. Only attacking and defending advice persists at all times.

EXISTING RESEARCH

Intelligent Tutoring systems

There are myriad research papers about ITS. The most authoritative statement about the effectiveness of different ITS seems to come from Steenbergen-Hu. [25] In his study of over 20 different ITS, he reported no difference between various IT. Other studies report similar results [23].

Ma investigated what makes an ITS effective. He reported that there was no specific explanation, but attributed the similarity in results to the fact that they all implement the important features. He said the most effective features are "individualized task selection, prompting and response feedback", and that these features are the most "frequently implemented" [17].

Chess research

Chess, although popular and widely studied, is usually not the subject of scholarly academic research. Most chess papers focus on the "perceptual advantages of experts over novices" [13]. Papers specifically on teaching and learning chess are scarce.

The paper most related to The Chess Tutor is *Automated chess tutor*, by Sadikov, Možina, Guid, Krivec, & Bratko. [24]. This paper's main focus is to turn chess analysis into plain English. They recruited chess experts to come up with about 25 hand-crafted chess concepts that they turn into code. They give examples such as:

```
if (BLACK_BISHOPS = 2) then if
(BLACK_BISHOPS_POSITION +
BLACK_BISHOPS_MOBILITY +
BLACK_BISHOPS_BLOCK_CENTER +
BLACK_BISHOP_PAIR +
BLACK_BISHOP_KING_SAFETY <= -65) then
comment("Black has an active bishop pair.")
```

This is not so much a tutor as an expert system. It 'reads' a chessboard and spits out which rules or concepts the board satisfies. It does not dispense advice nor change its analysis depending on the state of the game. The idea is similar to the internal workings of a chess engine, except the technical analysis (figure 2) is in plain English. This is basically how The Chess Tutor works except The Chess Tutor is geared towards dispensing advice and teaching chess high level concepts, not just analyzing a board.

The same authors try again in *Building an Intelligent Tutoring System for Chess Endgames* [11]. They embrace the Intelligent tutoring system model and use a rule-based or model-tracing intelligent tutoring system. In this paper they focus solely on the chess end game involving a king, bishop and knight versus a lone king (also known as KBNK – a particularly pernicious problem in chess circles). The paper is only four pages long and they do not get into specifics, but like their previous paper, they hand-code

certain rules and check for their completion. Unlike their previous work where they just scanned a static board state, in this project they scans the *moves* a player makes, and keeps score of how well that move fulfilled a specific concept. It has the excellent concept of skill bars which measure each skill, that grow or decrease depending on the players moves.

Figure 11 - figure from *Building an Intelligent Tutoring System for Chess Endgames*

The authors write several follow up papers based on these projects.

In Toward modeling task difficulty: the case of chess [14], the authors compare problem solving in chess with problem solving in life, and the ways a person breaks down a problem into its core elements.

In Learning positional features for annotating chess games: A case study [10], the authors again discuss the difficulty in coming up with the correct chess rules or concepts to model. They find some of the same problems this author discovered in the field of teaching chess, mainly that online chess tutoring are just tutorials, mostly on chess endgames.

In Fighting Knowledge Acquisition Bottleneck with Argument Based Machine Learning [18], the authors apply machine learning techniques to their chess rules. They use the chess rule 'bad bishop' and have experts say if this board indicates an instance of this rule. After getting enough yes/no examples, they feed the boards into a machine learning algorithm which then is able to predict if that board is an example of 'bad bishop', without human help. The results were not good, with only 59% correct classification accuracy.

In *Arguments in interactive machine learning*, the author discusses a theoretical combination of machine learning and expert guidance to come up with more accurate rules. [19]

Note that these papers are all by the same group of authors. They all discuss the difficulty in coming up with plain-English rules that govern chess. One cannot help but wonder why they do not mention existing chess engines such as Stockfish. The chess experts they recruited were undoubtedly proficient at the game of chess, but Stockfish is hundreds of well-established, proven to be effective, chess concepts. One wonders whether their reported difficulties arise because they created their own algorithms, instead of using proven chess engine algorithms.

Not all chess research involves the Guid & Mozina group. In *The benefits of chess for the intellectual and social-emotional enrichment in schoolchildren*, schoolchildren are divided into groups that play chess, and those that do not, and compare their development afterwards. [1] They found that chess improves cognitive abilities, coping and problem-solving capacity, and even socio-affective development of children and adolescents who practice it.

A good summary of chess technology and chess software is *Chess Software And Its Impact On Chess Players.* [8] It surveys some 250 chess applications and breaks them down by what concepts they taught and how (e.g. endgame, traps, patterns; exercises, tests, or studies).

Other research approaches chess from a more psychological point of view. *Training in chess: A scientific approach* attempts to break down and categorize chess knowledge. [12] It relies on the classic 'chunking' theory of chess [3] to break down chess knowledge into implicit vs explicit, theory vs strategy, and beginning, middle and endgame positions.

As a final honorable mention, *The impact of the search depth on chess playing strength* [9] is a wonderful work correlating rankings with search depth, and if The Chess Tutor ever incorporated chess ELO rankings, one of the most widely used metrics to measure a players chess ability, it would use this paper.

MOTIVATION AND EMOTION

Intelligent tutoring systems are not a magical replacement for a human tutor. They have not proven to be more effective at teaching than a human, and they suffer from a lack of motivation. Students usually profess boredom or frustration with their intelligent tutoring system, and quit the lesson or learn minimally. [2, 20]

The Chess Tutor attempts to overcome some of these challenges by using the latest research in ITS and keeping users interest. Specifically, The Chess Tutor incorporates avatars, and empathetic statements.

The first method The Chess Tutor employs to overcome boredom and improve learning is the use of Avatars. Avatars, or talking heads, provide a visual variety to the interface. (Or auditory variety! See [4]) But avatars do more than provide stimulus. Studies have shown that students ascribe emotions and personalities to avatars, and when advice or feedback comes from an avatar, and not an emotionless machine, students process or internalize the advice differently. Negative advice ("that move was wrong"), when given without an avatar, tends to make students feel frustrated and discouraged. However if that same advice is perceived to come from an avatar, the

students tend to displace their feelings onto the avatar, and feel that the avatar is frustrated or unhappy – not themselves. [2]

Due to this research, The Chess Tutor uses three different Avatars: a positive avatar (a happy pawn), a neutral avatar (a king), and a negative avatar (an angry knight). [21] Most messages will appear with the happy pawn avatar. If advice is more of a studious (or boring) nature, the king will appear. Negative statements appear using the angry knight avatar.

Figure 12- The Chess Tutor's avatars

The second important motivational concept The Chess Tutor employs is empathetic statements. There are many different ways to give advice in an ITS. Studies have shown that fact-based success/failure messages decreases motivation. Intrinsically motivated messages or 'growth mindset' messages such as 'it is important to keep trying and learn from mistakes' were also not effective. The best method is to use empathetic statements. [20] As a result of this research, The Chess Tutor attempts to phrase its advice in as friendly a tone as possible, even at the cost of simplicity or clarity.

Figure 13 - happy, empathetic advice

TECHNICAL DETAILS

The Chess Tutor was created in Html and JavaScript using all open-source software. The chessboard graphics are courtesy of chessboardjs.com. [7] The rules of the game and internal state of the game is done using chess.js. [6] The computer opponent is a JavaScript version of Stockfish, the world's leading chess engine. [15, 26]

As is usual when programming Intelligent Tutoring systems, each concept, each line of dialogue, and a response for each possible choice; must be hand coded into the system. There is no magical way to turn chess notation into high level concepts. The specific chess concepts encoded are: basic attack, basic defense, controlling the center, getting knights and bishops out, castling, and check. Some of the code, such as attacking and defending detection, is modifications of The Stockfish Evaluation Guide [27] – other code was created from scratch by the author.

The code consists of a main loop that is triggered when a piece is moved. Depending on whose turn it is, The Chess Tutor dispenses either advice or feedback. The different chess concepts are scanned and anything that applies is put into a master hint list, that a player can access one hint at a time by pressing the Hint button.

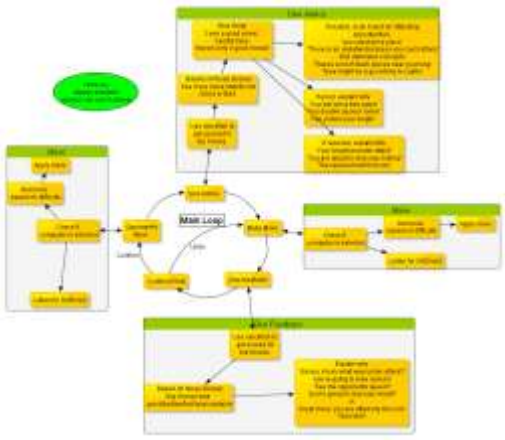

Figure 14 - a flow chart of The Chess Tutor's execution

ONLY A BEGINNING

The Chess Tutor is not a complete solution. Due to the complexity of encoding chess concepts and the limited time, only a few chess concepts were programmed into The Chess Tutor. The choice was made to program and teach the beginning, or opening, concepts of the game.

Creating a program that focuses on the beginning of the game is important for several reasons.

- End game lessons are abundant. There is no need to create another system that teaches more end-game concepts.
- Chess and its openings are difficult to learn, even at an intermediate level. Due to this difficulty, most players do not learn these beginning concepts, and there is almost always a need to improve beginning concepts.
- One of the greatest needs for intelligent tutoring systems are in the classrooms. Classrooms are almost by definition populated by younger players. Younger players are not likely to grasp more complicated chess concepts, and would benefit more from beginning chess concepts.
- Existing beginner chess lessons focus on extremely basic things like how the pieces move, or the rules of Castling. There is a need for more

abstract concepts that apply at the beginning of the game. A beginning player is separate from the beginnings, or openings, of a chess game.

Other creators of chess tutors are encouraged to focus their efforts on the beginning and middle of the game more than the end game.

THE FUTURE

The Chess Tutor demonstrates that it is possible to create a full-fledged tutor that teaches not just the basic concepts of chess, but all concepts, from the mid-game through to the final checkmate. These concepts must all be coded by hand, but it is doable.

Work has already been done in breaking down the highly-optimized Stockfish code into smaller chunks at The Stockfish Evaluation Guide. At The Stockfish Evaluation Guide one can browse the concepts on the left hand side and view each algorithm. The graph feature is highly illuminating at visually explaining how all these separate concepts come together to create a reliable measure of board strength, the heart of a chess engine.

The difficulties in creating chess rules, as discussed in papers by Hristova *et al.* ,[13, 14] are not intractable. It was once thought impossible to create a computer capable of playing chess, yet through years of plugging away at the problem, the chess programming community has created Stockfish – an open-source project that encodes every relevant chess concept into a blazing-fast algorithm that can beat anything, human or computer. The knowledge already exists. It is just a matter of rewriting it to accommodate human readability. Future improvements to The Chess Tutor would be breaking down these chess concepts into human readable forms.

CONCLUSION

The goal of The Chess Tutor is to demonstrate to the chess community that it is possible to teach high-level chess to players of all levels while playing a game of chess. Playing against the computer *can* be more than just practice. Learning and improving does not have to be a series of stand-alone static lessons, nor a post-game analysis in algebraic notation.

The Chess Tutor uniquely contributes to the field by showing that there can exist a standards-based format for teaching chess which has all the essential features of a human tutor in a chess classroom or club. The standard is an Intelligent Tutoring System which has been shown to be at least as effective as a human tutor, and emulates all the essential features of a human tutor, namely: playing an entire game of chess and intelligently getting adaptable advice in plain English. Previous efforts in this field have not emphasized the idea of teaching chess concepts while playing an entire game of chess

Chess concepts, while complex, are not mysterious. They do not need to be created by experts. Thanks to open source projects like the Stockfish Evaluation Guide [27], a non-chess programmer has the tools needed to turn chess concepts into human-readable, plain English advice. Projects such as The Chess Tutor demonstrate that it is possible to build on existing solutions to teach chess to anyone, at any level, worldwide.

REFERENCES

1. Aciego, R., García, L., & Betancort, M. 2012. The benefits of chess for the intellectual and social-emotional enrichment in schoolchildren. The Spanish journal of psychology, 15(2), 551-559.
2. Baylor, Amy L. 2009. "Promoting motivation with virtual agents and avatars: role of visual presence and appearance." Philosophical Transactions of the Royal Society B: Biological Sciences 364, no. 1535 (2009): 3559-3565.
3. Chase, William G., and Herbert A. Simon. 1973. "Perception in chess." Cognitive psychology 4, no. 1 (1973): 55-81.
4. Chess challenge. 1999. Anonymous. Popular Electronics, Vol.16(6), pp.22-24
5. Chess Free. 2019. www.snipca.com/12323. Retrieved April 2019
6. Chess.js. 2019. https://github.com/jhlywa/chess.js Retrieved April 2019
7. Chessboard.js. 2019. Chessboardjs.com. Retrieved April 2019
8. Dhou, K. 2008. Chess Software And Its Impact On Chess Players. https://doi.org/10.24124/2008/bpgub1368
9. Ferreira, Diogo R. 2013. "The impact of the search depth on chess playing strength." ICGA Journal 36, no. 2 (2013): 67-80.
10. Guid, M., Možina, M., Krivec, J., Sadikov, A., & Bratko, I. 2008. Learning positional features for annotating chess games: A case study. In International Conference on Computers and Games (pp. 192-204). Springer, Berlin, Heidelberg.
11. Guid, M., Mozina, M., Bohak, C., Sadikov, A., & Bratko, I. 2013. Building an Intelligent Tutoring System for Chess Endgames. In CSEDU (pp. 263-266).

12. Gobet, F., & Jansen, P. J. 2006. Training in chess: A scientific approach. Education and chess.

13. Hristova, Dayana, Matej Guid, and Ivan Bratko. 2014. "Assessing the difficulty of chess tactical problems." International Journal on Advances in Intelligent Systems 7, no. 3 (2014): 728-738.

14. Hristova, D., Guid, M., & Bratko, I. 2014. Toward modeling task difficulty: the case of chess. In COGNITIVE 2014, the sixth international conference on advanced cognitive technologies and applications (pp. 211-214).

15. Javascript version of Stockfish. 2019. https://www.npmjs.com/package/stockfish. Retrieved April 2019

16. Kodaganallur, V., Weitz, R. R., & Rosenthal, D. 2005. A comparison of model-tracing and constraint-based intelligent tutoring paradigms. International Journal of Artificial Intelligence in Education, 15(2), 117-144.

17. Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. 2014. Intelligent tutoring systems and learning out comes: A meta-analysis. Journal of Educational Psychology, 106(4), 901-918.

18. Možina, M., Guid, M., Krivec, J., Sadikov, A., & Bratko, I. 2008. Fighting Knowledge Acquisition Bottleneck with Argument Based Machine Learning. In ECAI (pp. 234-238).

19. Možina, Martin. 2018. "Arguments in Interactive Machine Learning." Informatica 42, no. 1.

20. Pekrun, Reinhard, Thomas Goetz, Lia M. Daniels, Robert H. Stupnisky, and Raymond P. Perry. 2010. "Boredom in achievement settings: Exploring control–value antecedents and performance outcomes of a neglected emotion." Journal of Educational Psychology 102, no. 3 (2010): 53

21. Rubner, R. 2019. Caricatures by Rex. https://www.facebook.com/Caricatures-by-Rex-174483932609130/. Retrieved April 2019

22. Wikipedia.org. 2019. Algebraic chess notation. https://en.wikipedia.org/wiki/Algebraic_notation_(chess). Retrieved April 2019

23. Sabo, K. E., Atkinson, R. K., Barrus, A. L., Joseph, S. S., & Perez, R. S. 2013. Searching for the two sigma advantage: Evaluating algebra intelligent tutors. Computers in Human Behavior, 29(4), 1833-1840.

24. Sadikov, A., Možina, M., Guid, M., Krivec, J. and Bratko, I., 2006, May. Automated chess tutor. In International Conference on Computers and Games (pp. 13-25). Springer, Berlin, Heidelberg.

25. Steenbergen-Hu, S., & Cooper, H. 2013. A meta-analysis of the effectiveness of intelligent tutoring systems on K–12 students' mathematical learning. Journal of Educational Psychology, 105(4), 970.

26. Stockfish. 2019. www.stockfishchess.org. Retrieved April 2019

27. The Stockfish Evaluation Guide. 2019. https://hxim.github.io/Stockfish-Evaluation-Guide/ Retrieved April 2019

APPENDIX

A video presentation of The Chess Tutor is at

https://youtu.be/4I3hgR2Ev4o

The code is at

https://github.gatech.edu/jktejik3/thechesstutor

The working product is at

www.thechesstutor.com