Issue 076: Chess

De Programmatica Ipsum

2025-01-06

Contents

Issue 076: Chess **1**

A Panegyric For Human Chess Players **3**

Bartek Spitza **15**

David Levy & Monty Newborn **19**

Issue 076: Chess

By Adrian Kosmaczewski, January 6th, 2025

Welcome to the 76th issue of *De Programmatica Ipsum*, about *Chess*.

In this edition:

- We explore the role of humans[1] in a world where computers are undeniably better at chess than us.

[1] https://deprogrammaticaipsum.com/a-panegyric-for-human-chess-players/

- In the Library section[2], we review "How Computers Play Chess" by David Levy and Monty Newborn[3].
- In our Vidéothèque section[4], we learn from Bartek Spitza[5] how to create a chess engine.

Download this issue in PDF[6] or EPUB[7] format, and read it in your preferred device.

We would like to thank our patrons who generously contribute every month (or have contributed in the past) to our work and help us run this magazine. Thank you so much! In alphabetical order: Adam Guest, Adrian Tineo Cabello, Benjamin Sheldon, Christopher Nascone, Colin Powell, Franz Lucien Moersdorf, Guillermo Ramos Álvarez, Jean-Paul de Vooght, Dr. Juande Santander-Vela, Patryk Matuszewski, Paul Hudson, Quico Moya, Roger Turner, Szymon Licau, and countless more leaving anonymous tips every month.

Enjoy this issue! Please subscribe to our free newsletter[8] to stay updated about new releases, share the articles on social media, or contribute[9] if you would like to support our work with a donation via Liberapay[10].

Cover photo by Randy Fath[11] on Unsplash[12].

---

[2] https://deprogrammaticaipsum.com/category/library/

[3] https://deprogrammaticaipsum.com/david-levy-monty-newborn/

[4] https://deprogrammaticaipsum.com/category/videotheque/

[5] https://deprogrammaticaipsum.com/bartek-spitza/

[6] https://deprogrammaticaipsum.com/pdf/issue-076-chess.pdf

[7] https://deprogrammaticaipsum.com/epub/issue-076-chess.epub

[8] https://deprogrammaticaipsum.com/newsletter/

[9] https://deprogrammaticaipsum.com/contribute/

[10] https://liberapay.com/akosma/donate

[11] https://unsplash.com/@randyfath?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

[12] https://unsplash.com/photos/selective-focus-photography-of-chess-pieces-G1yhU1Ej-9A?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

A Panegyric For Human Chess Players

By Adrian Kosmaczewski, January 6th, 2025

Carl Sagan[13], in the opening words of his 1977 bestseller book, "The Dragons of Eden", quotes a phrase from Plato[14]'s Phaedrus[15], saying "In good speaking, should not the mind of the speaker know the truth of the matter about which he is to speak?" In this occasion I ask myself the same question, and thus I start this article by stating openly

[13]https://deprogrammaticaipsum.com/carl-sagan/
[14]https://en.wikipedia.org/wiki/Plato
[15]https://en.wikipedia.org/wiki/Phaedrus_(dialogue)

that I am no expert in the game of chess (I actually think I have a negative Elo rating!) I am just a fascinated onlooker who happens to marvel at the intricacies of a game that I have never mastered, most probably never will, but one which I greatly enjoy watching and reading about.

For some weird reason we did not mention the game of chess in our article about gaming[16] two years ago; we have, however, used the allegory represented by its quintessential board as a heading picture in three previous articles of this magazine: the one by Graham about artificial intelligence[17] in our August 2019 issue, the title page[18] of our Management issue or June 2021, and the one about Alan Perlis[19] of September 2022.

Despite these clichés, chess deserves a separate issue of its own, and yet another misuse of header images with chessboards and knights. Not because we would like to debate whether it is a sport or not[20] (we think it is), or whether because we would love to mention the names of past and present legendary chess players (we will, anyway), but because chess is at the core of the history of computer programming, and it has had (and continues to have) a timeless appeal to generations of computer scientists.

Not only that, but also because we probably owe the existence of an entire field of research called "artificial intelligence" to the humble game of chess.

In the first chapter of his 1989 book "The Emperor's New Mind"[21], British Nobel Laureate Sir Roger Penrose[22] asked the billion-dollar question: "Can a Computer Have a Mind?"

> Chess-playing computers probably provide the best examples of machines exhibiting what might be thought of as 'intelligent behaviour'. In fact, some machines have now (in 1989) reached an extremely respectable level of performance in relation to human players – approaching that of 'International Master'. (These computers' ratings would be a little below 2300, where, for comparison, Kasparov, the world champion, has a rating greater than 2700.)
> (...)
> (This difference is even more noticeable with the difficult Oriental game

[16] https://deprogrammaticaipsum.com/insert-coin/
[17] https://deprogrammaticaipsum.com/artificial-intelligence-bias-and-opportunity/
[18] https://deprogrammaticaipsum.com/issue-33-management/
[19] https://deprogrammaticaipsum.com/alan-perlis-and-the-evolution-of-programming-languages/
[20] https://www.chessjournal.com/is-chess-an-olympic-sport/
[21] https://en.wikipedia.org/wiki/The_Emperor%27s_New_Mind
[22] https://en.wikipedia.org/wiki/Roger_Penrose

> of 'go', where the number of possibilities per move is considerably greater than in chess.)

The dream of a machine playing chess is literally as old as the French Revolution. The Mechanical Turk[23] (not to be confused with Amazon's exploitation marketplace of the same name[24]), later immortalized[25] by Edgar Allan Poe, dazzled naïve kings and businessmen from 1770 to 1854 on both sides of the North Atlantic. Then came Ajeeb[26], who did the same from 1868 to the 1920s, following the same design pattern as its predecessor: a concealed human player inside a box. At the end of the nineteenth century, Mephisto[27] was an electromechanical device, operated at a certain distance by a human being, a dramatic effect for unsuspecting audiences between 1868 and 1889.

The first recorded true chess automaton, but often ignored by many, was "El Ajedrecista"[28], built in 1912 by Leonardo Torres Quevedo in Madrid, Spain. This machine was an actual, independent electromechanical device, and it could play an endgame situation, consisting of moving a white king and a rook to checkmate a human player escaping defeat with a black king. To add praise to its wondrous design, El Ajedrecista could even signal illegal moves, and was the first machine to beat a Grandmaster (in this case, the Polish Savielly Tartakower[29].)

The birth of computing brought a new impetus to the long-standing dream of a machine playing chess. As explained[30] by Silver, Hubert, Schrittwieser et al. in their 2018 paper about AlphaZero,

> The study of computer chess is as old as computer science itself. Charles Babbage, Alan Turing, Claude Shannon, and John von Neumann devised hardware, algorithms and theory to analyse and play the game of chess.

They conveniently forgot to mention Konrad Zuse, however. As mentioned by Graham in a previous article[31] of this magazine,

> The Z3 was the first Turing-complete machine, and Zuse went on to write

[23] https://en.wikipedia.org/wiki/Mechanical_Turk
[24] https://www.mturk.com/
[25] https://en.wikipedia.org/wiki/Maelzel's_Chess_Player
[26] https://en.wikipedia.org/wiki/Ajeeb
[27] https://en.wikipedia.org/wiki/Mephisto_(automaton)
[28] https://en.wikipedia.org/wiki/El_Ajedrecista
[29] https://en.wikipedia.org/wiki/Savielly_Tartakower
[30] https://www.science.org/doi/10.1126/science.aar6404
[31] https://deprogrammaticaipsum.com/on-research-software-engineering/

> the first computer chess program in the first high-level programming language Plankalkül, a language of his own devising and an unacknowledged forerunner to ALGOL. This was all part of his PhD thesis, but having failed to pay the submission fee to the University of Augsburg he did not obtain a degree.

Later, John von Neumann[32] and Oskar Morgenstern published in 1944 "Theory of Games and Economic Behavior", a book which features a section (15.7) called "Application to Chess".

> Examples of games with perfect information were already given in 6.4.1.: Chess (without chance moves) and Backgammon (with chance moves). Thus we have established for all these games the existence of a definite value (of a play) and of a definite best strategies.

Claude Shannon[33] wrote an article in 1950 for "The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science", called "Programming a Computer for Playing Chess"[34], where he explains:

> The chess machine is an ideal one to start with, since: (1) the problem is sharply defined both in allowed operations (the moves) and in the ultimate goal (checkmate); (2) it is neither so simple as to be trivial nor too difficult for satisfactory solution; (3) chess is generally considered to require "thinking" for skilful (sic) play; a solution of this problem will force us either to admit the possibility of a mechanized thinking or to further restrict our concept of "thinking"; (4) the discrete structure of chess fits well into the digital nature of modern computers.

According to Shannon, the thinking of von Neumann and Morgenstern effectively defines an algorithm for a machine to play chess:

> These two facts imply (von Neumann and Morgenstern, 1944) that any given position of the chess pieces must be either:
>
> (1) A won position for White. That is, White can force a win, however Black defends.
>
> (2) A draw position. White can force at least a draw, however Black

[32] https://deprogrammaticaipsum.com/william-aspray/
[33] https://en.wikipedia.org/wiki/Claude_Shannon
[34] https://www.tandfonline.com/doi/abs/10.1080/14786445008521796

> plays, and likewise Black can force at least a draw, however White plays. If both sides play correctly the game will end in a draw.
>
> (3) A won position for Black. Black can force a win, however White plays.
>
> This is, for practical purposes, of the nature of an existence theorem.

Shannon is aware, however, of the staggering amount of calculation such an algorithm would entail:

> However, even at this figure there will be 10120 variations to be calculated from the initial position. A machine operating at the rate of one variation per micro-second would require over 1090 years to calculate the first move!

Not very practical indeed. But Shannon defined in this paper the two major requirements for a software program that plays chess:

1. A data structure representing the board.
2. An evaluation function to determine the best next movement.

Those are, in a nutshell, the core elements of pretty much every chess program. The first requirement is a relatively trivial problem; the second, not so much, and humans have followed two major strategies to solve it:

1. Hard-coded algorithms.
2. Machine learning.

Even if we, earthlings of the 21st century, would like to believe that the second option is relatively recent, history shows that both strategies grew up somewhat in parallel, until a series of AI winters[35] stalled all progress between the 1970s to the 2010s.

Let us go back in time. An article[36] in the July 1959 edition of the IBM Journal by Arthur Lee Samuel[37] proposed and popularized the concept of machine learning: "Some Studies in Machine Learning Using the Game of Checkers". But why checkers? Well, the answer is quite simple, really:

> A game provides a convenient vehicle for such study as contrasted with a problem taken from life, since many of the complications of detail are

[35] https://en.wikipedia.org/wiki/AI_winter
[36] https://ieeexplore.ieee.org/document/5392560
[37] https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist)

> removed. Checkers, rather than chess, was chosen because the simplicity of its rules permits greater emphasis to be placed on learning techniques.

"Samuel Checkers" as it is known, ran on an IBM 704[38], and it is the first time a machine learning system based on a reward function reached "a respectable amateur status", according to Wikipedia[39], playing thousands of games against itself in order to learn the game, and this until the mid-1970s!

We will come back to the machine learning strategy; because between the 1970s to the 2000s, the reigning strategy was the purely algorithmic one, slowly growing in power thanks to Moore's law.

During the 1970s and 1980s, in parallel to the rise of the personal computer, the gaming market was invaded by chess machines, almost forgotten small portable computers that could only play chess. The website "Chess Computer UK"[40] celebrates and showcases them, many of which were quite popular even in my native and crisis-ridden Argentina. The Mephisto[41] line of chess computers deserves a special mention in this category, created by a Swiss company called Saitek[42] that was later bought by the also Swiss-born Logitech.

The aforementioned "Chess Computer UK" website also contains a section with an archive of articles[43] published in various magazines, like our beloved Byte, from the late 1970s to 1986. Who can forget Robert Tinney[44]'s iconic cover for the October 1978 issue[45] of Byte Magazine dedicated to the subject of "Chess for the Microcomputer"?

Another interesting tidbit: did you know that Microchess[46] was the one of the first successful commercial software packages in the history of personal computing? Released in 1976 for the KIM-1 microcomputer at a price of 10 dollars (around 55 USD in 2025), and later ported to other classics of the era such as the Apple II, the Commodore PET, or the TRS-80, it generated millions of dollars of revenue until the mid-90s, despite being (according to chess experts) a relatively poor implementation.

---

[38] https://deprogrammaticaipsum.com/ken-ross-paul-laughton/
[39] https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist)
[40] http://chesscomputeruk.com/
[41] https://en.wikipedia.org/wiki/Mephisto_(chess_computer)
[42] https://en.wikipedia.org/wiki/Saitek
[43] http://chesscomputeruk.com/html/publication_archive_1974-79.html
[44] https://en.wikipedia.org/wiki/Robert_Tinney
[45] https://archive.org/details/byte-magazine-1978-10
[46] https://en.wikipedia.org/wiki/Microchess

(The revenue generated by Microchess financed the development of VisiCalc, another landmark in the history of software. But I digress. Again.)

Appropriately enough, the secret plan by IBM[47] to build the 5150 Personal Computer[48] was called "Project Chess". Speaking about PC software games, who can forget wonderful classics such as Chessmaster 2000[49] or Rebel[50]?

(By the way, the actor featured in the cover of Chessmaster 2000, Will Hare[51], passed away in 1997, a few months after Deep Blue defeated Kasparov.)

Speaking about which: the road to Deep Blue was long, and its history is better told by its creators themselves, in a 2002 paper[52] by Murray Campbell, A. Joseph Hoane, and Feng-hsiung Hsu:

> Earlier efforts in building a chess machine, ChipTest and Deep Thought, took place at Carnegie Mellon University in the 1980s. In 1988 Deep Thought was the first chess machine to beat a Grandmaster in tournament play.
> (...) Deep Blue defeated Garry Kasparov in the 1997 match by a score of 3.5–2.5. For this victory, the Deep Blue team was awarded the Fredkin prize for defeating the human world champion in a regulation match.

Boom. The tipping moment of centuries of research in chess-playing machines was, without a doubt, the defeat of Garry Kasparov by Deep Blue[53] in 1997:

> Deep Blue derived its chess prowess through brute force computing power. It used 32 processors to perform a set of coordinated, high-speed computations in parallel. Deep Blue was able to evaluate 200 million chess positions per second, achieving a processing speed of 11.38 billion floating-point operations per second, or flops.

That was 11.38 gigaflops of brute force. For comparison, the computer leading the TOP500 list[54] at the time of this publication, "El Capitan", is able a peak performance

[47] https://deprogrammaticaipsum.com/think/
[48] https://deprogrammaticaipsum.com/peter-norton/
[49] https://en.wikipedia.org/wiki/Chessmaster_2000
[50] https://rebel13.nl/
[51] https://en.wikipedia.org/wiki/Will_Hare
[52] https://doi.org/10.1016/S0004-3702(01)00129-1.
[53] https://www.ibm.com/history/deep-blue
[54] https://top500.org/lists/top500/2024/11/

of... 2.746.30 **peta**flops, or 2.7463 × 106 gigaflops. On top of the Green500 list[55], the JEDI system (love the name) performs at almost 73 gigaflops per watt (!)

Needless to say, the Apple Silicon CPU[56] you are probably using to read this magazine is definitely[57] a worthy opponent for your next game. Hang tight.

To make a really long history really short, less than 20 years after Deep Blue, AlphaZero[58] fulfilled the dream of a "Samuel Chess" once and for all: a neural network able to play chess at an unimaginable level of mastery, but without any brute force.

In lieu of a Mechanical Turk we got a Digital Turk instead, and quite a powerful one at that.

To summarize, we are in 2025, merely 8 years after AlphaGo, 27 years after Deep Blue, 65 years after "Samuel Checkers", 112 years after El Ajedrecista, 254 years after the Mechanical Turk, and we have long left behind the time when humans could win against a machine playing chess. It is endgame[59], people.

(Which begs the question: what could be the next step? Quantum computers playing quantum chess? And against whom?)

These days you do not need to code your own chess engine from scratch anymore. There are libraries for Python[60], C[61], Go[62], Rust[63], Java[64] (here is another one[65]), JavaScript[66], WASM[67], C++[68], C#[69], and countless more in pretty much every programming language you can imagine, ready to be included in your own software.

Of course, if you are still interested in writing your own chess engine, there is an *ad*

[55] https://www.top500.org/lists/green500/2024/11/
[56] https://deprogrammaticaipsum.com/eternally-finally/
[57] https://arxiv.org/pdf/2211.00720
[58] https://www.science.org/doi/10.1126/science.aar6404
[59] https://www.computerhistory.org/chess/endgame/
[60] https://pypi.org/project/chess/
[61] https://www.gnu.org/software/chess/
[62] https://github.com/notnil/chess
[63] https://docs.rs/chess/latest/chess/
[64] https://github.com/megamiii/JavaChessGame
[65] https://github.com/LeeStephen/Chess
[66] https://github.com/josefjadrny/js-chess-engine
[67] https://github.com/nmrugg/stockfish.js/
[68] https://github.com/official-stockfish/Stockfish
[69] https://github.com/lynx-chess/Lynx

hoc wiki[70], a page in the Computer History Museum[71], a whole section of Wikipedia[72], and a free e-book[73] for that.

Finally, if you are not a programmer but are looking to challenge a suitable computer, there is GNU Chess[74], Leela Chess Zero[75], Stockfish[76], Lichess[77], Chess.com[78], GNOME Chess[79], Fritz[80], macOS Chess[81], Crafty[82], Rybka[83], and Maia[84], all ready to help you get better at the game. Oh, and the GNOME Chess Clock[85] could be useful, too.

(By the way, did you know you could not remove[86] the Chess.app application from macOS a few years ago? Also, it has been revamped early this year[87], have you seen it?)

Speaking about getting better at the game, remember to save them for future reference in files using the PGN format[88] to ensure portability; GNOME Chess and Chess.com both support it already.

At the beginning of Steven Spielberg's (and, to a certain degree, also Stanley Kubrick's) 2001 movie "Artificial Intelligence"[89], a certain Professor Hobby (played by the late William Hurt) says[90]:

> To create an artificial being has been the dream of man... since the birth

---

[70] https://www.chessprogramming.org/Main_Page
[71] https://www.computerhistory.org/chess/
[72] https://en.wikipedia.org/wiki/Category:Computer_chess
[73] https://www.adamberent.com/wp-content/uploads/2019/02/GuideToProgrammingChessEngine.pdf
[74] https://en.wikipedia.org/wiki/GNU_Chess
[75] https://lczero.org/
[76] https://stockfishchess.org/
[77] https://lichess.org/
[78] https://www.chess.com/
[79] https://wiki.gnome.org/Apps/Chess
[80] https://en.wikipedia.org/wiki/Fritz_(chess)
[81] https://support.apple.com/guide/chess/welcome/mac
[82] https://en.wikipedia.org/wiki/Crafty
[83] http://rybkachess.com/
[84] https://www.maiachess.com/
[85] https://apps.gnome.org/en/Chessclock/
[86] https://www.reddit.com/r/mac/comments/c3hrsg/i_didnt_realize_chess_was_vital_to_macos/
[87] https://9to5mac.com/2024/06/14/macos-sequoia-apple-chess-game/
[88] https://en.wikipedia.org/wiki/Portable_Game_Notation
[89] https://www.imdb.com/title/tt0212720/
[90] https://www.imdb.com/title/tt0212720/characters/nm0000458

> of science. Not merely the beginning of the modern age... when our forbearers astonished the world with the first thinking machines: primitive monsters that could play chess.

The silver lining of living in the future is that we can take back chess for our own human pleasure; that of the meeting of minds, the silent battle, the surprising challenge. Just us, between fellow organic members of the human race, sharing a delicate and exclusive moment, trying to outsmart each other. It is not about Elo rankings or being the next Miguel Najdorf[91] (creator of the eponymous variation[92] of the Sicilian defense) or the next Faustino Oro[93]; it is about being surprised, about learning new strategies, and about sharing that strong handshake at the end of a tough match.

We humans should let machines play and replay millions of times per second those historical matches in their databases, like the Polish Immortal[94]. Instead, we should focus in enjoying the present moment, as the pieces move through the board, unfolding new and unforeseen combinations. If possible, we should avoid playing chess with pigeons[95], if you see what I mean.

As a personal anecdote, I remember my *professeur de mathématiques*[96] Gilbert Elia organizing chess tournaments at the Collège Sismondi[97] where I finished high school in the early nineties. He would ask me to join, and to my pitiful answer "I don't know how to play" he would just jokingly reply "you've got one hour to learn and come join us".

Now that I think about it, I should have done exactly that. Not only for the game itself, but to have shared that moment with those people in that place.

Cover photo by Mick De Paola[98] on Unsplash[99], showing the oversized chessboards and Staunton[100]-inspired pieces in Geneva's Parc des Bastions[101], where I often met

[91] https://en.wikipedia.org/wiki/Miguel_Najdorf

[92] https://en.wikipedia.org/wiki/Sicilian_Defence,_Najdorf_Variation

[93] https://en.wikipedia.org/wiki/Faustino_Oro

[94] https://en.wikipedia.org/wiki/Polish_Immortal

[95] https://rationalwiki.org/wiki/Pigeon_chess

[96] https://deprogrammaticaipsum.com/issue-55-mathematics/

[97] https://en.wikipedia.org/wiki/Coll%C3%A8ge_Sismondi

[98] https://unsplash.com/@mickdepaola?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

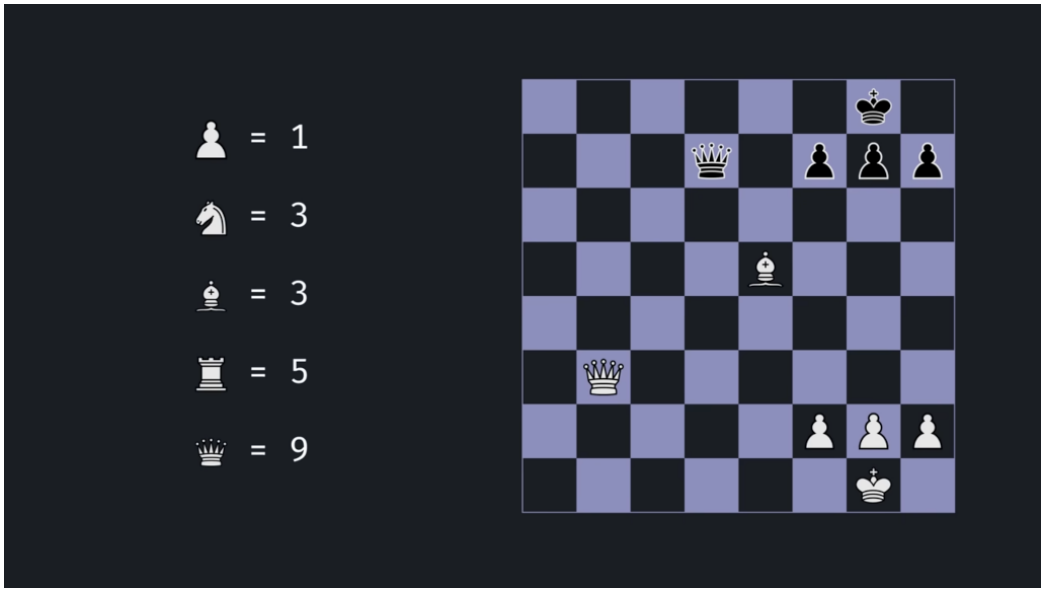
[99] https://unsplash.com/photos/a-man-standing-in-front-of-a-giant-chess-set-YqJFSr2-DiQ?utm_content=creditCopyText&utm_medium=referral&utm_source=unsplash

[100] https://en.wikipedia.org/wiki/Staunton_chess_set

[101] https://fr.wikipedia.org/wiki/Parc_des_Bastions

professeur Elia playing chess on Sunday afternoons.

Bartek Spitza



By Adrian Kosmaczewski, January 6th, 2025

Truth or dare! I defy any of my readers to refute this simple fact: watching "The Queen's Gambit"[102] on Netflix during the pandemic made you to either dust out that old chessboard in the attic, or sign up for a chess.com[103] account. I know I did both things, and not only because Anya Taylor-Joy[104] grew up in literally the same neighborhood of Buenos Aires where I did.

Do you know what other Argentine did actually start playing chess during the pan-

[102]https://www.imdb.com/title/tt10048342/
[103]https://www.chess.com/
[104]https://en.wikipedia.org/wiki/Anya_Taylor-Joy

demic, but with much more success than most of us? A kid named Faustino Oro[105], born in Buenos Aires in 2013. Yes, at the time of this publication he is barely 11 years old, and has been recently named the youngest ever international chess master in history, with an Elo rating of 2433. True to this wonder, less than 4 years after he moved a pawn for the first time, he beat up the one and only chess grandmaster Magnus Carlsen[106], with a rating of 2831.

Talk about a successful Netflix show! Congrats to them for this. Because, you know, making a TV show about chess is not easy, let alone one with wide critic acclaim. If you do not believe me, ask the poor producer at the BBC who ran a short-lived and interactive chess program on national British TV. Yes, interactive: people in the audience could call live 0898 99-11-99 and suggest the next move against a grandmaster in the studio. When I say it was short-lived, I mean it: it aired just once, on December 7th, 1990. You can watch the broadcast on YouTube[107] nowadays, and it is... well, let us say it is a far cry from The Graham Norton Show[108], if you see what I mean.

(The BBC had already by 1990 a somewhat long tradition around the chess-on-TV concept: some of my British readers might remember "The Master Game", a program dedicated to the game of chess which aired from 1975 to 1983, and also available on YouTube[109] for nostalgia purposes.)

But I digress. The issue at stake is that chess, as interesting as it is, and as passionate its hardcore audience may be, does not have the liveliness of other sports like, say, football or Formula 1. Explaining chess on video, particularly for educational purposes, can quickly become a tedious experience for the watcher.

Let alone explaining how to program a computer that plays chess. The risk of alienation through sheer boredom is the highest ever.

This is why we have chosen Bartek Spitza's "The Fascinating Programming of a Chess Engine"[110] as this month's Vidéothèque entry. Bartek[111] is a software developer from Gothenburg, Sweden, who created Sophia, a UCI[112]-compatible chess engine written

[105] https://en.wikipedia.org/wiki/Faustino_Oro
[106] https://en.wikipedia.org/wiki/Magnus_Carlsen
[107] https://www.youtube.com/watch?v=QIOwwGIupp0
[108] https://en.wikipedia.org/wiki/The_Graham_Norton_Show
[109] https://www.youtube.com/@BBCTheMasterGame
[110] https://www.youtube.com/watch?v=w4FFX_otR-4
[111] https://www.linkedin.com/in/bartek-spitza/
[112] https://en.wikipedia.org/wiki/Universal_Chess_Interface

in C and available on GitHub[113].

In this video, Bartek explains the two basic elements that make up a chess engine: first, the data structures required to store the positions of pieces on the board. (Spoiler alert: 64-bit integers have exactly the same number of bits as a chessboard has squares.) Second, the algorithm that takes a current chessboard and returns the best possible next move. For this second element, the video dives in a step-by-step explanation of its algorithm, unsurprisingly choosing Python[114] as a quintessential tool for teaching[115] the concepts behind.

His video is remarkable not only for the minimalistic yet powerful animations, or by the calm tone of his voice (giving an almost ASMR[116] feeling to the video). The explanations of the data structures and the minimax[117] algorithm in use are simple and to the point.

Along the same lines, Bartek has also published a short video[118] explaining how the Stockfish chess engine works, and it is totally worth a watch, too.

I wish to Bartek to continue explaining technical subjects, and similarly, I urge all of my readers to subscribe to his channel and support his work with the usual shares, likes, and other positive signs of encouragement. His most recent video[119] (at the time of this writing) explains how text encoding works, while the first one[120] he published in 2023 dealt with the inner workings of the Java[121] virtual machine. Always with the same sober visual style, and with excellent explanations to follow along.

This month's Vidéothèque video is "The Fascinating Programming of a Chess Engine" by Bartek Spitza, and you can watch it on YouTube[122]. Complement it with "Creating a Chess AI with TensorFlow"[123], because we live in the 21st century and machine learning is all the rage these days.

And while we wait for Anya to reprise the role of Beth Harmon in a new season of

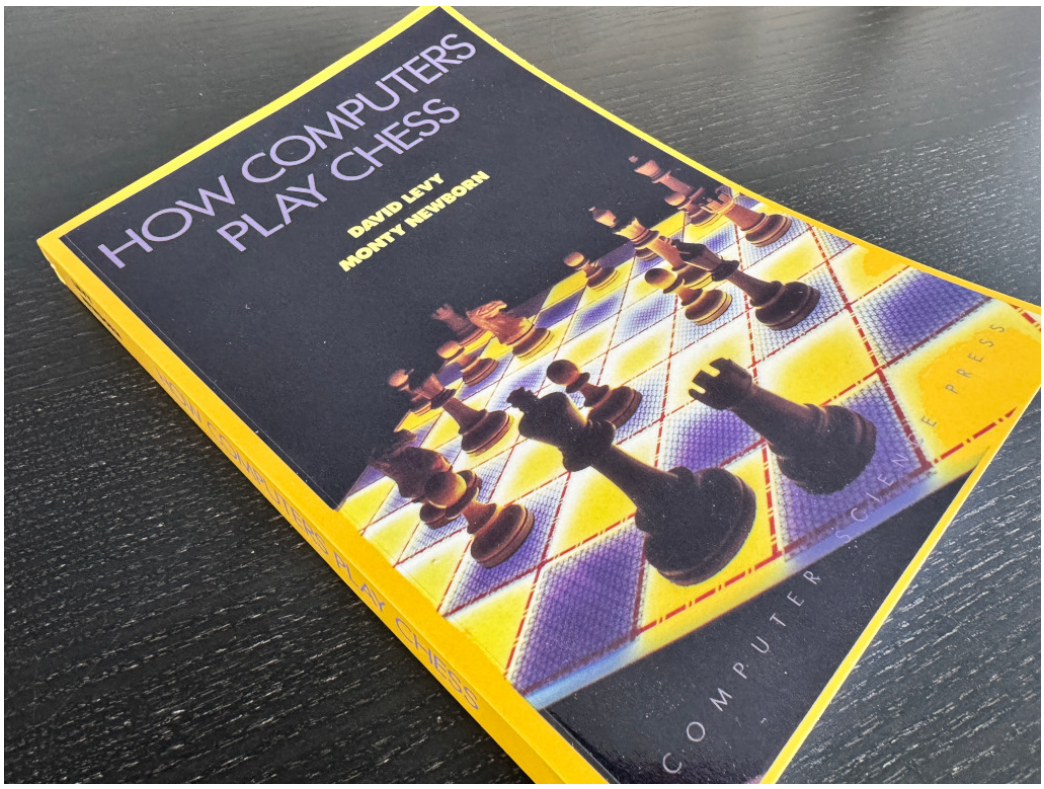
[113]https://github.com/bartekspitza/sophia
[114]https://deprogrammaticaipsum.com/the-state-of-python-in-2021/
[115]https://deprogrammaticaipsum.com/banning-adopting-reckoning/
[116]https://en.wikipedia.org/wiki/ASMR
[117]https://en.wikipedia.org/wiki/Minimax
[118]https://www.youtube.com/shorts/YRYi_LLrUdY
[119]https://www.youtube.com/watch?v=xAihdJLlmPw
[120]https://www.youtube.com/watch?v=zJPFwGs4q9o
[121]https://deprogrammaticaipsum.com/write-anywhere-run-once/
[122]https://www.youtube.com/watch?v=w4FFX_otR-4
[123]https://www.youtube.com/watch?v=ffzvhe97J4Q

"The Queen's Gambit", we can watch "Rematch"[124], a series produced by the French-German TV channel Arte[125] recreating the Kasparov versus Deep Blue match of 1997.

Cover snapshot chosen by the author.

[124]https://www.imdb.com/title/tt26741906/
[125]https://en.wikipedia.org/wiki/Arte

David Levy & Monty Newborn



By Adrian Kosmaczewski, January 6th, 2025

The literature about and around chess is too long to enumerate in an article of a thousand words, and this is clearly not my intent. If you want to learn chess, there are so many good books around, it is hard to pick just one. There are, however, fewer books explaining the art of how to teach a computer to play chess; and the one chosen for this

issue of the Library section does a magnificent job at precisely that.

The name of David Levy[126] might not ring a bell immediately, but there cannot be any quick incursion in the world of chess programming without it popping up, and repeatedly: after all, he has written more than 40 books about the subject of chess computer programming.

The Library book of this month is one of those works: "How Computers Play Chess"[127], published in 1991 and written by David Levy and Monty Newborn[128].

The late 80s were an interesting moment for computer chess. Garry Kasparov[129], who reigned as world chess champion from 1985 to 2000, was busy defending the human crown against a cohort of ever more powerful computers by IBM. This *invictus* status would eventually disappear[130] in 1997. But back in 1989, the potential of a computer beating a human being in the sacrosanct game of chess was still a conjecture, at best.

For Levy and Newborn, the stakes were clear, and the title of the first chapter of the book says it all: "The Challenge is World Champion Kasparov". Said chapter describes in detail the match between a first iteration of a chess supercomputer by IBM, the less well-known "Deep Thought"[131]. It was a strong contender, having defeated quite a few grandmasters along the way (including the aforementioned Levy), but was no match for Kasparov in August 1989. The same team, led by ACM Grace Murray Hopper Award winner Feng-hsiung Hsu[132], would strike back with Deep Blue less than seven years later.

The book goes on to describe the history of chess computers, going as back as Babbage, and following with early efforts by Zuse and Shannon. The evolution of chess game engines is studied with examples from both sides of the Berlin Wall[133].

(Kids: remember that this book was written around the final times of the Cold War[134], and there were geopolitical arguments at play in every field of human knowledge, including chess and artificial intelligence. If you do not believe me, just read about the

[126] https://www.chessprogramming.org/David_Levy
[127] https://archive.org/details/howcomputersplay0000levy
[128] https://en.wikipedia.org/wiki/Monty_Newborn
[129] https://en.wikipedia.org/wiki/Garry_Kasparov
[130] https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov
[131] https://en.wikipedia.org/wiki/Deep_Thought_(chess_computer)
[132] https://en.wikipedia.org/wiki/Feng-hsiung_Hsu
[133] https://en.wikipedia.org/wiki/Berlin_Wall
[134] https://en.wikipedia.org/wiki/Cold_War

highly publicized game of Boris Spassky versus Bobby Fischer in Reykjavík in 1972[135] for proof.)

After this historical introduction, the book dives into more and more interesting implementation details: opening and endgame databases, coding strategies, search algorithms, and even an evaluation of current (as of 1991) chess programs and software packages on the market.

There is, however, a fabulous thing about this book: each famous match between humans and computers, in every generation of chess engines, is illustrated by its corresponding sequence of moves in standard notation. The idea being that, to understand the growing capabilities of chess engines, there is no better way than to see how good (or bad!) these engines could play at every stage of history, from the late 1950s to the early 1990s.

(Spoiler alert: the almost exponential increase in ability of those engines was nothing short of outstanding.)

It is through this progression that we discover the evolution of chess engines; and at the same time, we can grasp some sense of the subjective evaluation of good and bad moves (typically highlighted with `!` or `??` signs throughout the text, as is usually done), and ironically enough, learn some good chess strategies along the way.

Both Levy and Newborn know what they are talking about. David Levy is a skilled player, who has regularly challenged the most powerful chess programs of their time: after winning against Chess 4.7[136] in 1978, and against the powerful Cray Blitz[137] for the Cray[138] supercomputer in 1984[139], he was defeated[140] by Deep Thought in 1989. On the other hand, Monty Newborn is the former chairman of the Computer Chess Committee of the Association for Computing Machinery (ACM) and a professor of computer science at McGill University in Montreal.

After the blunder and defeat[141] of Russian grandmaster Vladimir Kramnik[142] against

[135] https://en.wikipedia.org/wiki/World_Chess_Championship_1972
[136] https://www.chessprogramming.org/Levy_versus_Chess_1978
[137] https://en.wikipedia.org/wiki/Cray_Blitz
[138] https://en.wikipedia.org/wiki/Cray
[139] https://www.chessprogramming.org/Advances_in_Computer_Chess_4#LevyCrayBlitz
[140] https://www.chessprogramming.org/Levy_versus_Deep_Thought_1989
[141] https://en.wikipedia.org/wiki/Blunder_(chess)#Deep_Fritz_vs._Vladimir_Kramnik
[142] https://en.wikipedia.org/wiki/Vladimir_Kramnik

Deep Fritz[143] in 2006, an article on the New York Times[144] quoted:

> Today's outcome may end the interest in future chess matches between human champions and computers, according to Monty Newborn, a professor of computer science at McGill University in Montreal. Professor Newborn, who helped organize the match between Mr. Kasparov and Deep Blue, said of future matches: "I don't know what one could get out of it at this point. The science is done."

It is worth noting that Kramnik was well-versed in anti-computer tactics[145] to win the game, as he had shown a few years earlier[146], but that was not enough in 2006. According to Wikipedia[147],

> Newborn was belatedly correct; as of 2021, that was the last serious attempt by a world-class player to defeat a top chess machine/program.

Endgame[148], really.

The world has changed a lot since the publication of "How Computers Play Chess". Kasparov has been beaten by Deep Blue. IBM went on to win at Jeopardy![149]. And Lee Sedol[150] has lost[151] against AlphaGo[152]. This is why Dr. Levy has shifted his attention to other problems brought by artificial intelligence, like the possibility of marrying robots[153] in Massachusetts in 2050, and other crunchy (and NSFW)[154] subjects. Dr. Levy likes to bet about the future, clearly[155].

If you are still interested in the latest developments around game engine programming, here go two more recommendations from a sea[156] of extraordinary[157] books: "Guide

[143]https://en.wikipedia.org/wiki/Fritz_(chess)
[144]https://www.nytimes.com/2006/12/05/crosswords/chess/05cnd-chess.html
[145]https://en.wikipedia.org/wiki/Anti-computer_tactics
[146]https://en.wikipedia.org/wiki/Brains_in_Bahrain
[147]https://en.wikipedia.org/wiki/Monty_Newborn
[148]https://www.computerhistory.org/chess/endgame/
[149]https://en.wikipedia.org/wiki/IBM_Watson
[150]https://en.wikipedia.org/wiki/Lee_Sedol
[151]https://en.wikipedia.org/wiki/AlphaGo_versus_Lee_Sedol
[152]https://en.wikipedia.org/wiki/AlphaGo
[153]https://www.nbcnews.com/id/wbna21271545
[154]https://en.wikipedia.org/wiki/Love_and_Sex_with_Robots
[155]https://www.chess.com/blog/Ginger_GM/the-history-of-computer-chess-part-5-levys-bet
[156]https://www.chessprogramming.org/Recommended_Reading
[157]https://www.lkessler.com/ccbooks.shtml

to Programming a Chess Engine"[158] by Adam Berent, reflecting his own step-by-step process of creating a chess engine in C# from 2008 to 2016. And then, akin to our modern times, "Neural Networks for Chess: The magic of deep and reinforcement learning revealed"[159], by Dominik Klein (2022), explaining the inner workings of new, machine learning-based engines such as AlphaZero[160], Leela Chess Zero[161], and Stockfish NNUE[162], together with an interesting comparison with brute-force approaches such as that of Deep Blue.

Is programming your own chess engine a way to gain some mastery in the game? Yes and no; it would make wonders to your programming skills, no doubt about that; but I do believe that a healthy and regular dose of chess challenges with actual human beings, preferably in front of a physical chessboard, would have a healthier effect in your psyche. We are social beings, after all; let us not forget that aspect, and let us cultivate it, too.

Of course, if what you want are books about learning how to play chess, I do have some recommendations, starting with "Bobby Fischer Teaches Chess"[163], one of the biggest bestsellers of its genre, by none other than one of the greatest chess players[164] of the 20th century. I can also heartily recommend "How to Win At Chess: The Ultimate Guide for Beginners and Beyond"[165] by Levy Rozman, a recent but very interesting addition to my chess bookshelf, written by a very well-known and popular streamer[166]. Finally, I can also recommend at least one book from the extensive written works of legendary champion José Raúl Capablanca[167]; in this case, "Arte y Secretos del Ajedrez"[168], a wonderful and relatively short introduction to the game, in Spanish this time.

This month's Library book is "How Computers Play Chess"[169] by David Levy and Monty Newborn, and remember: it is best read with a real chessboard by your side.

---

[158]https://www.adamberent.com/wp-content/uploads/2019/02/GuideToProgrammingChessEngine.pdf

[159]https://arxiv.org/pdf/2209.01506

[160]https://en.wikipedia.org/wiki/AlphaZero

[161]https://en.wikipedia.org/wiki/Leela_Chess_Zero

[162]https://www.chessprogramming.org/Stockfish_NNUE

[163]https://en.wikipedia.org/wiki/Bobby_Fischer_Teaches_Chess

[164]https://en.wikipedia.org/wiki/Bobby_Fischer

[165]https://chess.co.uk/collections/endgame-strategy/products/how-to-win-at-chess-the-ultimate-guide-for-beginners-and-beyond-levy-rozman

[166]https://www.gothamchess.com/

[167]https://en.wikipedia.org/wiki/Jos%C3%A9_Ra%C3%BAl_Capablanca

[168]https://www.casassaylorenzo.com/Papel/9789505900268/ARTE+Y+SECRETOS+DEL+AJEDREZ

[169]https://archive.org/details/howcomputersplay0000levy

Cover photo by the author.