Candidate Number: 1020694

A Checkered Past: Game-Playing in Artificial Intelligence, 1945 – 1973

MSc History of Science Technology and Medicine

Trinity Term 2019

Word Count: 16,486

Abstract

Game-playing has played a fundamental role in the history of computing. Electronic computers were invented for military use during World War II, but they were quickly made to play games like draughts and chess. This thesis explores who, when, and why scientists were programming computers to play games in the UK between 1945 and 1973, and ties game-playing to the larger political context in this period. Programming non-numerical games required a substantial number of conceptual and practical innovations, including the invention of programming languages, machine learning, and search algorithms. The individuals who pioneered these techniques—principally Alan Turing, Christopher Strachey, Donald Michie, and Arthur Samuel, for the purpose of this thesis—used game-playing as a launching pad to hone their craft, and their contributions have subsequently been recognized as essential to computing as a whole.

Chess also served as a potent metaphor for the Cold War, and the same mathematical techniques that solved chess endgames were used to make decisions on the international stage. Governments invested deeply in the scientists who worked on draughts and chess, with the expectation that those games would yield military-ready innovations; the scientists in turn consulted for military and government organizations. Over time, game-playing became a benchmark for the state of AI as a whole, and researchers became increasingly overconfident in the prospects of their research. In 1973, Sir James Lighthill wrote a report questioning the usefulness of AI as a discipline, leading to large research funding cuts in British universities. I will show that the failure of chess programs to meet their promise was a prominent feature of his critique, and ultimately set AI research back in the UK by a decade.

Acknowledgements

This thesis was made possible by the immensely helpful guidance of my supervisor, Jacob Ward. I would not come across half of these sources, nor thought so deeply about the topic at hand, had it not been for your recommendations and advice. Thank you as well to Chris Hollings, who supported this project from the beginning. I am also very grateful to the History of Science, Medicine, and Technology department who patiently tutted, but never more, waiting for my late assignments.

I have the best writing partner imaginable in David Laing. At this point, I don't need to send him my documents for editing; we've done this so many times that his voice pops up in my head automatically. This thesis would be considerably worse had it not been for the help of Kay MacDonald. She made more speedy, useful, droll, and valuable suggestions than I can even count, and validated my concerns about the mismatch between myself and my discipline. Alice Campbell Davis was physically adjacent to me for the majority of the writing of this document, and tolerated complaints, groans, spontaneous interjections about esoteric 1950s computer programmers, rhythmless dancing, and blatant violations of household norms. She is a creative, a flavor architect, and the funny one.

I am immensely appreciative of Pamela McCorduck, who met with me for three hours to discuss the ideas and people she first wrote about in 1979. I hope I can do her work justice in some small way. I also must thank Richard Rhodes, Douglas Hofstadter, and Steven Pinker, whose writing has most strongly influenced my own, and whose style I am continuously failing to replicate successfully.

Funding for my archival visit to Stanford University was provided by St. John's College.

Table of Contents

Introduction: More Than a Game

In the summer of 1956, twenty scientists assembled at Dartmouth University to inaugurate a new academic discipline.[1] It was called "artificial intelligence." They gathered to discuss how electronic computers, which had been invented barely a decade earlier, could be used to understand language, form concepts, solve human problems, and self-improve.[2] The attendees included celebrated mathematicians, engineers, computer scientists, economists, and physicists. But aside from their mutual interest in intelligent machines, these scientists shared something unexpected in common: programming computers to play games. Of the twenty people who attended the Dartmouth workshop, at least seven had written a program to play chess, draughts, or noughts-and-crosses.[3]

Game-playing has been tied to computing since its earliest days. We can trace it all the way back to Charles Babbage, the irascible Victorian mathematician whose crank- and gear-based Differential and Analytical Engines predated modern computers by a century. Babbage wanted to finance the construction of his machines—which were never actually assembled in his lifetime—and concluded that it would be profitable to build "a machine that should be able to play a game of purely intellectual skill successfully; such as tit-tat-to [*sic*], drafts, chess, &c".[4] Babbage dismissed the possibility after some consideration, on account of the "myriads of combinations which even the simplest games included".[5]

Babbage's story is a mirror to the 20th century. Early computing pioneers relied on game-playing for funding, viewed games like draughts and chess as a peak intellectual endeavor, and struggled mightily to build machines that could wade through "myriads of combinations" of moves. To these scientists, like the ones at the 1956 Dartmouth

---

[1] Ronald Kline, "Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence." *IEEE Annals of the History of Computing* 33, no. 4 (2011), 5.
[2] John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence: August 31, 1955." *AI Magazine* 27, no. 4 (2006), 12
[3] Ray Solomonof, "Who plus Aug 21-25," 1956, The Dartmouth AI Archives, Box B, http://raysolomonoff.com/dartmouth/boxbdart/dart56ray812825who.pdf.
[4] Charles Babbage, *Passages from the Life of a Philosopher*. London: Longman, Green, Longman, Roberts, & Green, 465.
[5] Ibid., 466

Conference, game-playing was not frivolous. According to one of the earliest books on artificial intelligence (AI), "[game playing] provides a direct contest between man's wit and machine's wit . . . In short, game environments are very useful task environments for studying the nature and structure of complex problem-solving processes".[6] Game-playing was a window into cognition, but also a window into a wider world. Early computing cannot be separated from its Cold War context, which itself was viewed as an elaborate game between two symmetric adversaries. Computers which were programmed to solve chess endgames were equally tasked with computing nuclear missile trajectories and avoiding doomsday scenarios.

My central thesis is that game-playing served a foundational role in the history of computing. My argument is two-fold. First, I demonstrate that games like chess and draughts were an almost obsessive focus for many leading computer scientists, and became the benchmark for assessing progress in AI, ultimately to the detriment of the field. Second, I show that games were inseparable from the military uses of computers. Wartime codebreakers were chess players, and draughts programmers were contracted by the Department of Defense. I will show how many fundamental innovations—programming languages, machine learning, reinforcement learning, hash tables—were developed trying to get computers to play games. My games of choice are chess and draughts (or "checkers" in American English), though I also briefly discuss the mathematical game Nim and noughts-and-crosses (or tic-tac-toe). Much has already been written about the American pioneers in this story—Marvin Minsky, John McCarthy, Herb Simon, and Allen Newell, to name a few—so my focus is on the UK, though the wartime story is an international one.

I begin with a survey of the historical literature on AI. Chapter 1 begins where computing begins, at Bletchley Park, home of the World War II codebreaking effort. The team at Bletchley was assembled around a group of chess-players, and was led by Alan Turing, an avid, albeit unimpressive, chess player in his own right. Less well-known but no less important was Donald Michie, a classicist-turned-codebreaker-turned-geneticist-turned-AI pioneer who will play a central

[6] Edward A. Feigenbaum & Julian Feldman, ed., *Computers and Thought*, (New York: McGraw-Hill Book Company, 1963).

role in this thesis. Chapter 2 turns to draughts, a much simpler game to program then chess. I discuss schoolteacher Christopher Strachey, whose foray into draughts influenced an American, Arthur Samuel, who used the game to write the first machine learning program. Chapter 3 starts by exploring the relationship between chess and nuclear war. I outline some minor successes in chess programming throughout the 1950s, including at Los Alamos, where the atomic bomb was built. Over-optimism took hold, however, and promises of chess supremacy went unfulfilled. AI soon faced massive budget cuts at the hand of Sir James Lighthill, whose infamous 1973 report is the focus of Chapter 4.

As the role of AI in 21st century life expands, it is important to reflect on the roots of the discipline. I have chosen to investigate the era between World War II and James Lighthill's 1973 report because it is close enough to the present to be recognizable and represents a period of immense and rapid change. With game playing as a lens, this thesis reflects on how academic fields become entranced by certain topics, how science and society cannot be disentangled, and how influential ideas are born.

Literature Review

Secondary Sources

The history of artificial intelligence is a small but growing niche within the history of science. Until the last decade, the history of AI has been written largely by its practitioners, with minimal input from professional historians. The first major history of the field is Pamela McCorduck's *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*.[7] McCorduck was married to computer scientist Joseph Traub, and the book is told through the lens of her interest in the field and her friendships with its practitioners. It serves as an essential oral history of AI 1940s through 1970s. McCorduck's book was rejected by 33 different publishing houses, but has since become a quasi-official history of the field's early days.[8,9] McCorduck's follow-up with computer scientist Ed Feigenbaum, *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World* paints a useful picture of the reinvigoration of computer science a decade after the Lighthill Report.[10]

Other major works of history have been written by AI practitioners, and have similarly involved interviews with major figures in the field. Computer scientists Nils Nilsson's *The Quest for Artificial Intelligence: A History of Ideas and Achievements* (2010) and Daniel Crevier's *AI: The Tumultuous History* provide sweeping overviews of achievements in the field. Another useful resource has been Rich Sutton's history of reinforcement learning, the field he revived in the 1980s.[11] These works present an internalist view of AI, focusing on technical breakthroughs and giving little attention to social, political, or economic factors that informed the types of problems AI researchers

---

[7] Pamela McCorduck, *Machines Who Think: A Personal Journey into the History and Prospects of Artificial Intelligence* (Massachusetts, AK Peters Ltd, 2004, 2ed), xiii.
[8] McCorduck, *Machines Who Think*, xiii.
[9] Jonnie Penn. "AI thinks like a corporation—and that's worrying." *The Economist*, November 2018.
[10] Pamela McCorduck and Ed Feigenbaum, *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World* (Reading: Addison-Wesley, 1983).
[11] Rich Sutton, "History of Reinforcement Learning," *Incomplete Ideas Blog*, 1 April 2005, http://www.incompleteideas.net/book/ebook/node12.html

worked to solve. They are useful, however, for understanding technical concepts like alpha-beta pruning and reinforcement learning.

Herb Simon was a keen documenter of his own field. Simon and checkers programmer Jonathan Schaeffer wrote a history of computer chess for a chapter in a 1992 edited volume.[12] Their collaboration is of historiographic interest in its own right, because Schaeffer went on to mathematically solve the game of checkers, a task completely unrelated to human cognition, while Simon was an economist who was mainly interested in computers insofar as they could illuminate and automate human decision-making.[13] Jonathan Schaeffer's book *One Jump Ahead* was also extremely valuable for its discussion of checkers, in particular its detailed analysis of the public response to Arthur Samuel's checkers program.[14]

Straddling the line between technical and social history is Andrew Hodges' exceptional biography *Alan Turing: The Enigma*.[15] Hodges is a mathematician, but his book paints a comprehensive portrait of Turing's life and work. It features prominently in my section on the origins of computer chess at Bletchley Park. In a similar vein is mathematician-cum-historian William Aspray's *John Von Neumann and the Origins of Modern Computing,* which was helpful for understanding the birth of the stored-program computer.[16]

In the 1970s and 80s, historians of science cared little for AI and computer science broadly, choosing to study the history of more respectable sciences like physics and mathematics. Later, when AI proved to be a groundbreaking science, an unnamed historian confessed that "nobody knew whether it would be important".[17]

[12] Herbert A. Simon & Jonathan Schaeffer, "Chapter 1: The Game of Chess" in *Handbook of Game Theory With Economic Applications*, vol 1 (1992).
[13] See e.g. Stephanie Dick, "Of Models and Machines: Implementing Bounded Rationality." *Isis* vol. 106, no. 3 (2015).
[14] Jonathan Schaeffer, *One Jump Ahead: Computer Perfection at Checkers* (Springer, 2009, 2ed).
[15] Andrew Hodges, *Alan Turing: The Enigma*, 2ed. (New Jersey: Princeton University Press, 2014).
[16] William Aspray, *John Von Neumann and the Origins of Modern Computing* (Cambridge: MIT Press, 1990).
[17] McCorduck, *Machines Who Think*, xiii.

We now know that AI is important, and historians have begun to catch up. The first analyses of artificial intelligence were enabled by the shifts that took place within the history and sociology of science in the 1970s, which emboldened scholars to critically appraise conventional narratives about technology.[18] Most relevant here is the book *Artificial Experts* by Harry Collins, one of the leaders of the "strong programme" of the sociology of scientific knowledge.[19] This was followed by the work of sociologist Mike Olazaran, who questioned the standard narrative about the discovery of neural networks and perceptrons in the 1950s-80s.[20] While my thesis does not discuss this controversy in depth, Olazaran's approach persuaded me that the stories we tell about AI often do not reflect reality.

Most of the traditional histories of AI ignore the political context around the field. The phrase "Cold War" appears precisely zero times in Nillson and McCorduck's books, for instance. To fill this gap, I drew from two important sources. The first is Paul Edwards' influential *The Closed World* (1996).[21] Edwards draws parallels between computing, cognitive psychology, and cybernetics, which arose from, and contributed to, a Cold War discourse of command and control. Edwards explores how metaphors linked humans and machines: memory, perception, thought, communication, and power all became human-computer "cyborg" concepts. Edwards discusses in particular the rise of artificial intelligence, which was supported heavily by the military establishment's drive toward automation. Edwards' book contributes heavily to my discussion of the relationship between chess and the Cold War, and inspired my investigation into the unexpected military associations of Christopher Strachey, Arthur Samuel, and Donald Michie.

---

[18] Works in this vein include: David Bloor, *Knowledge and Social Imagery* (Chicago: University of Chicago Press, 1976) which introduced the "strong programme"; Bruno Latour & Steve Woolgar, *Laboratory Life: The Social Construction of Scientific Facts* (Beverly Hills: Sage,1979);

[19] Harry Collins, *Artificial Experts* (Cambridge: MIT Press, 1990)

[20] Mikel Olazaran, "A Sociological Study of the Official History of the Perceptrons Controversy." *Social Studies of Science*, vol 26 (1996).

[21] Paul Edwards, *The Closed World: Computers and the Politics of Discourse in Cold War America* (Cambridge: MIT Press, 1996).

The second key recourse for the relationship between computing and the Cold War was Erickson et al.'s *How Reason Almost Lost its Mind*.[22] The book identifies a unique pattern of thought which they call "Cold War rationality": the belief that real-world problems can be modelled mathematically and solved algorithmically, that intelligence can be boiled down to symbolic processing, and that humans behave according to rational rules. This book was indispensable for my analysis of the minimax algorithm and the ubiquity of game theory in chess, war, and evolutionary biology.

The past decade has seen a boom in deep learning and artificial intelligence, and with it a rise in research dedicated specifically to the history of AI.[23] Stephanie K. Dick's path-breaking work on automatic theorem proving demonstrated how computers reshaped the notion of mathematical proof.[24] Its discussion of theorem-proving competitions, and the continuity of theorem proving from the 1950s to the present influenced my discussion on the origins and present state of computer chess. Recent work has also pointed out the need for more in-depth histories of machine learning.[25] I hope that this thesis represents a small step in that direction.

Another important thread is cybernetics. Cybernetics was an intellectual tradition founded by American mathematician Norbert Wiener to study "control and communication in the animal and the machine".[26] Cybernetics is about viewing minds as machines, and predated artificial intelligence, which is concerned with turning machines into minds.[27] Relevant histories of cybernetics include Geoff Bowker's "How to be Universal",[28] which demonstrates how cyberneticians gained legitimacy through a common esoteric language,

---

[22] Paul Erickson et al., *How Reason Almost Lost its Mind: The Strange Career of Cold War Rationality.* (Chicago: University of Chicago Press, 2013).

[23]

[24] Stephanie K. Dick, "After Math: (Re)configuring Minds, Proof, and Computing in the Postwar United States" PhD dissertation, Harvard University, Graduate School of Arts & Sciences, 2014.

[25] Aaron Plasek, "On The Cruelty of Really Writing a History of Machine Learning." *IEEE Annals of the History of Computing*, vol 38(4) (2016).

[26] Norbert Wiener, *Cybernetics: Or Control and Communication in the Animal and the Machine*, 2ed. (Cambridge: MIT Press, 1961)

[27] Edwards, *The Closed World*, 239.

[28] Geoff Bowker, "How to be Universal: Some Cybernetic Strategies, 1943-70." *Social Studies of Science*., vol 23 (1993), 108.

and Ronald Kline's "Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence",[29] which provides a useful description of AI's founding.

All of the work discussed up to now forms the backbone for this thesis, but two papers in particular form a growing body of literature to which I am consciously adding. The first is Nathan Ensmenger's indispensable "Is chess the drosophila of artificial intelligence? A social history of an algorithm"[30] The paper explores how a focus on chess shaped and ultimately misdirected AI research in the 1970s, and situates chess games within a broader Cold War context. The aim of my paper is similar, though I am focusing equally on draughts and more intensely on the contributions of Christopher Strachey, Arthur Samuel, and Donald Michie. The second formative work is Jon Agar's unpublished "What is science for? The Lighthill report and the purpose of artificial intelligence research." I will build on Agar's analysis by focusing specifically on Lighthill's critiques of chess.

Finally, I am indebted to the Chess Programming Wiki, run by chess programmer Mark Lefler.[31] It has pointed me to many of the papers I've already discussed, and helped structure my research.

Primary Sources

My main primary sources are from two sets of archives. The first is the Christopher Strachey archive in the Bodleian Library at Oxford University. The second is the John McCarthy Papers at Stanford Libraries. My principal information about Donald Michie came from reading his correspondences with Strachey and McCarthy. I also visited the Royal Institution archives in London for resources on the Lighthill debate.

I made extensive use of two books which document in detail the early days of AI in the US and the UK. The first is BV Bowden's *Faster Than Thought*, which was the first

---

[29] Ronald Kline, "Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence,"
[30] Nathan Ensmenger, "Is chess the drosophila of artificial intelligence? A social history of an algorithm." *Social Studies of Science*, vol 42 no. 1 (2011):5-30
[31] Mark Lefler, *Chessprogramming Wiki*, 2019, https://www.chessprogramming.org/Main_Page

popular account of computing in the UK.[32] The section written by Alan Turing and Christopher Strachey on game-playing is particularly valuable. Next is Ed Feigenbaum and Julian Feldman's *Computers and Thought*, which is a compendium of various papers which were presented at the 1956 Dartmouth workshop.[33] Additionally useful were the conference proceedings from the 1958 Mechanisation of Thought Processes conference in Teddington.[34]

Alan Turing's writings at the National Physical Laboratory were valuable for the history of computers, and are conveniently collected online by Jack Copeland.[35] The mathematical and computing literature on chess-playing begins with Shannon (1950)[36]. Articles by Bernstein (1958)[37], Newell, Simon, & Shaw (1958)[38], and Samuel (1959)[39] round out the decade. The 1960s open with Donald Michie and John Maynard Smith's "Machines that Play Games".[40] The academic literature beyond this point turns extremely technical, and since this is not a history of heuristic search algorithms, my use of those resources ends there. That being said, Donald Michie did continue to publish interesting articles on chess well into the 1970s.[41]

For my analysis of Michie and McCarthy's bet with David Levy, I made use of the Computer History Museum's Oral History interview with Levy.[42] And finally, I relied on the

---

[32] BV Bowden, ed. *Faster Than Thought: A Symposium on Digital Computing Machines*. (London: Sir Isaac Pitman & Sons Ltd., 1953).
[33] Edward A. Feigenbaum & Julian Feldman, ed., *Computers and Thought*, (New York: McGraw-Hill Book Company, 1963).
[34] National Physical Laboratory. *Mechanisation of Thought Processes: Proceedings of a Symposium Held at the National Physical Laboratory on 24th - 27th November 1958*. (No. 10. London: H.M.S.O, 1959).
[35] Jack Copeland, "Catalogue: Turing's Automatic Computing Engine," 2019, http://www.alanturing.net/turing_archive/archive/index/aceindex.html
[36] Claude Shannon, "Programming a Computer for Playing Chess," *The London, Ediburgh, and Dublin Philosophical Magazine and Journal of Science*, 41 no. 304 (1950): 256-275
[37] Alex Bernstein & Michael de V. Roberts, "Computer v. Chess-Player", *Scientific American*, vol. 198 no. 6, June 1958
[38] Allen Newell, J.C. Shaw, & Herbert A. Simon, "Chess-Playing Programs and the Problem of Complexity," *IBM Journal of Research and Development*, vol 2(4) (1958):320-335.
[39] Arthur Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol 3(3) (1959): 210-229.
[40] John Maynard Smith and Donald Michie, "Machines That Play Games," *The New Scientist* no. 260, 9 November 1961
[41] Examples include: Donald Michie "Programmer's Gambit," *New Scientist*, 17 August 1972; Donald Michie, "Machines and the Theory of Intelligence," *Nature*, vol. 241 (1973): 507-512.
[42] Computer History Museum, "Oral History of David Levy,", 8 September 2005, https://www.computerhistory.org/chess/orl-4345632d88ad1/

Computing At Chilton website for hosting the Lighthill Report, the replies to that report, and the video of the 1973 Lighthill debate on the BBC series *Controversy*.[43]

With this background set, I now begin to explore the history of game-playing in computing by asking: why were computer scientists so entranced by chess?

[43] Computing At Chilton, "Lighthill Report,"

Chapter 1: The Touchstone of Intellect?

Origins of an Obsession

The pioneers of computing in the 1940s and 1950s came from diverse intellectual traditions. Some, like Stanford mathematician John McCarthy and English mathematician and wartime codebreaker Alan Turing, were interested in logic and problem-solving. Others were cyberneticians, like information theorist Claude Shannon and mathematician Norbert Wiener, who wanted to model the brain using concepts from engineering and information theory.[44] Still others, like economist Herb Simon, were interested in using machines to model human decision making for applications in business. Arthur Samuel and Alex Bernstein were corporate engineers who were not employed to conduct original research. Sanislaw Ulam was a nuclear physicist who helped usher in a thermonuclear bomb. They all had one thing in common: chess.

Chess captivated the imagination of nearly every major scientist interested in AI. For Herb Simon and his colleagues Allen Newell and J.C. Shaw, automating chess meant "penetrating the core of human intellectual endeavour".[45] For Ed Feigenbaum and Julian Feldman, editors of the 1963 compendium *Computers and Thought,* which featured the papers presented at the Dartmouth conference, chess is "one of man's valued intellectual diversions".[46] It is "the intellectual game *par excellence*",[47] "one of the most sophisticated human activities",[48] and the *sine qua non* of cognition.[49] Ironically, however, none of the founders of AI were particularly good it. Turning was by all accounts a poor chess player, though he did compensate for his inadequacies by inventing a variant of chess where the players had to run around the garden between moves.[50] Norbert Wiener was similarly inept. Cognitive scientist Marvin Minsky recalls Wiener sitting in the faculty club at MIT

---

[44] Paul Edwards, *The Closed World*, 240.
[45] Newell, Shaw, and Simon, "Chess-Playing Programs and the Problem of Complexity", 320.
[46] Feigenbaum & Feldman, *Computers and Thought*, 37.
[47] Ibid., 39.
[48] Bernstein and Roberts, "Computer v. Chess-Player" 2.
[49] Pamela McCorduck, personal communications, 24 July 2019.
[50] McCorduck, *Machines Who Think*, 66.

frustratedly losing every game he initiated.[51] Plenty of laudatory words have been written about John McCarthy and Herb Simon's chess programs, but none about their chess playing. And conversely, there is no reason to believe that chess players are any smarter than anyone else outside of chess.

Why, then, was chess such an important benchmark for judging progress in computing? What in the game captivated the minds of these men so profoundly? There are a number of plausible answers. Chess was popular in intellectual break-rooms at universities, think tanks (most notably the RAND corporation, which is where Newell and Simon were based), and Oxbridge college common rooms.[52] Chess has long been a prestigious activity, associated with intellectuals, artists, and geniuses.[53] Mastery at chess was viewed as high-status in these settings, so perhaps mastery of chess programming would be viewed the same way.

Another reason is that, since chess was so difficult, it might yield fruitful new techniques in programming and computer design. The first person to express this idea in public was Claude Shannon. He wrote that solving chess will "act as a wedge in attacking other problems of a similar nature and of greater significance".[54] Some of the similar problems he mentions are machine translation, military decision-making, and musical composition. Similarly, Turing remarked in a 1953 edited volume called *Faster Than Thought* that "research into the techniques of programming a machine to tackle complicated problems of this type may in fact lead to quite important advances, and help in serious work in business and economics—perhaps, regrettably, even in the theory of war".[55] As we will soon see, the link between chess and war is in fact a strong one. Turing's thoughts were echoed by the editor of *Faster Than Thought*, B.V. Bowden, who spoke a great deal about game-playing in a 1961 speech to the British Association for the Advancement of Science. Bowden describes the facility with which computers can be made to play chess and draughts, and asks whether anything will "remain for them to conquer"

[51] McCorduck, *Machines Who Think*, 61.
[52] Erickson et al., *How Reason Almost lost is Mind*, 14.
[53] Ensmenger, "Is chess the drosophila of artificial intelligence?", 18
[54] Shannon, "Programming a Computer for Playing Chess", 256
[55] Turing in Bowden, ed., *Faster than Thought*, 285

after they've mastered game-playing.[56] The answer: business. Meetings, management, and budgeting can be made easier with computers, until "perhaps some day most of the work of the Board will no longer concern human beings at all".[57] Game-playing, then, is a gateway into automating larger areas of activity and work.

A final reason chess was so important "the sheer fun of the thing", to again quote Turing.[58] Most of the chess programmers were chess enthusiasts first, regardless of their skill level. Chess is ubiquitous, international, endlessly fascinating, and there are always new things to learn. It is no surprise, then, that it played such an important role in computing.

With some understanding of why chess loomed large in early computing, I will now turn to the history of computer chess itself. I will show that chess was deeply entwined with the codebreaking effort at Bletchley Park, and that chess computing was inseparable from its wartime roots.

Cracking the Chess Code

Whatever the motivation, the roots of computer chess run deep. The dream of building a machine that could play chess like a human is considerably older than computers themselves. In the 18th century, Hungarian engineer Wolfgang von Kempelen's "Mechanical Turk," a mechanical automation that purported to play chess without human intervention, wowed audiences around Europe.[59] It was eventually proven to be an elaborate hoax—a small man hid under the board and was obscured by an elaborate arrangement of mirrors—but not before playing against such distinguished opponents as Napoleon Bonaparte, Benjamin Franklin, and even Charles Babbage.[60]

---

[56] B.V. Bowden, "Discourse on 'The Impact of Automation'", *123rd Annuam Meeting of the British Association for the Advancement of Science*, 4 September 1961 Christopher Strachey Papers, MS. Eng. misc. b. 297/H.6
[57] Ibid.
[58] Turing in Bowden, ed., *Faster Than Thought*, 285
[59] William Clark, Jan Golinski, and Simon Schaffer. *The Sciences in Enlightened Europe*. (Chicago: University of Chicago Press, 1999), 158-160.
[60] Tom Standage, *The Turk: The Life and Times of a Famous Eighteenth-Century Chess-Playing Machine* (Berkeley: Berkeley Books, 2003), 140.

The Turk would go on to influence Babbage's later ideas about what his Analytical Engine might be capable of. As we have seen, Babbage wanted a machine which could "play a game of purely intellectual skill successfully; such as tic-tac-to, drafts, chess, &c".[61] Babbage had the idea of building a noughts-and-crosses or chess playing version of the Analytical Engine to sow interest in his work.[62]

Imitations of the Turk soon followed, but it took until the 1910s for the arrival of a genuine automated chess-playing machine. It was Spanish engineer Leonardo Torres y Quevedo's *El Ajedrescista*, a simple electromechanical endgame-player which could checkmate an opponent's king using a king and a rook.[63] For forty years, this device would remain the most sophisticated automated chess machine. The task of building anything more complex was simply too daunting.

But that did not stop people from discussing what it would take to automate chess in principle. One of the first people to consider the problem seriously was the Hungarian polymath John von Neumann. While studying the theory of games in the early 1920s, he realized that, in a game like chess, each player's optimal strategy must include the recognition that their opponent is trying to pursue an optimal strategy too.[64] In other words, each player must do the best they can possibly do while operating under the constraint that their opponent is trying to make them as worse of as possible. Thus "minimax" was born.[65] In the 1930s, Von Neumann gave regular talks on two-person zero-sum games. Alan Turing, who was then working on his PhD at Princeton, likely would have attended one of those lectures.[66]

Turing returned to his native England in 1938, and within a year he found himself working for the Government Code and Cypher School at Bletchley Park, leading a team tasked with the decryption of the German Enigma machine. Chess remained curiously

---

[61]Babbage, *Passages from the Life of a Philosopher*, 465.
[62] Ibid., 465
[63] McCorduck, *Machines Who Think*, 60.
[64] Aspray, *John von Neumann and the Origins of Modern Computing*, 15.
[65] Specifically, von Neumann's mathematical Minimax theorem proved that for any finite two-person zero-sum game, like chess or draughts, there exists an optimal strategy; see Erickson et al., *How Reason Almost Lost its Mind*, 138.
[66] Ibid., 268

close-by. One of the first people hired to work with Turing was the national chess champion and mathematician Hugh Alexander.[67] Soon after, another international player and grandmaster, Harry Golombek was directed away from his original assignment in the Royal Artillery and brought onto the cryptography team because of his chess ability.[68] Still another master player, Philip Milner-Berry, worked on the Enigma. They all played chess frequently during the codebreaking effort. Alexander led a beginner's chess course in the lab, and boards could be seen lying around during rare breaks from work.[69] Turing and fellow mathematician I. J. Good stayed up late discussing whether there was a definite method for playing chess, and how one might go about "solving" the game entirely.[70]

The connection between their codebreaking work and their chess prowess was not lost on these men. Golombek commented that codebreaking was "like playing chess, in that it involved getting into your opponent's head and imagining what he was thinking as he was developing or using a cipher".[71] Milner-Berry similarly said that code-breaking "was rather like playing a tournament game (sometimes several games) every day for five and a half years".[72]

Yet another new recruit at Bletchley was Donald Michie, an Oxford undergraduate classicist who became a teenage wartime cryptographer almost by accident.[73] One summer, he signed up for a Japanese language course for intelligence officers, but upon learning that the class was full ended up learning cryptography.[74] Within a few weeks he was at Bletchley. He was, like Turing, a wholly mediocre chess player, but had a keen interest in the game. Michie went on to play a fundamental role at Bletchley, aiding in the construction of a vacuum-tube-operated codebreaking device, the Colossus.[75]

---

[67] Ibid., 249.
[68] Ibid., 286.
[69] Hodges, *Alan Turing: The Enigma*, 260.
[70] Ibid., 267
[71] William D. Wall, "Chess World War II Codebteakers," 2018, http://billwall.phpwebhosting.com/articles/chess_codebreakers.htm
[72] Harry Golombek and Bill Hartston, *The Best Games of C.H.O'D Alexander* (London: Oxford University Press, 1976), 5
[73] Ibid., 292
[74] Stephen Muggleton, "Donald Michie," *The Guardian,* 10 July 2007. https://www.theguardian.com/science/2007/jul/10/uk.obituaries1.
[75] Hodges, *Alan Turing: The Enigma*, 379.

The Colossus was a fully electronic computational machine that could be programmed using a massive floor-to-ceiling plugboard.[76] Turing understood that such a device was not merely a useful tool for calculations. Rather, it represented a broader ideal: the possibility of a universal computation device.[77] By "universal," he meant the following: "If one can explain quite unambiguously . . . how a calculation is to be done, then it is always possible to programme any digital computer to do that calculation".[78] For Turing, a computer was not merely "a tool for reaching some end; it was . . . the embodiment of the possibilities and limits of logical thought".[79] The first use Turing imagined for such a universal device was chess.

Universal Turing Chess Machine

After the war, Turing took up a post at the National Physical Laboratory in London to build machines for the British Armed forces.[80] In 1945, he wrote a *Proposal for the Development in the Mathematics Division of an Automatic Computing Engine (ACE)*, published internally at the NPL and not released publicly until the 1980s.[81] His proposal contains a list of ten potential applications for the machine, to persuade the government of the project's usefulness. Item number one is standard for computers of the era: ballistic missile range tables.[82] But item number ten is far more interesting to Turing himself:

> Given a position in chess the machine could be made to list all the "winning combinations" to a depth of about three moves on either side. This is not unlike the previous problem, but raises the question, "Can the machine play chess?"[83]

---

[76] B. Jack Copeland, *Colossus: The Secrets of Bletchley Park's Codebreaking Computers*. (Oxford: Oxford University Press, 2006), 1.
[77] Hodges, *Alan Turing: The Enigma*, 368.
[78] Turing in Bowden, ed., *Faster Than Thought*, 289.
[79] Eloína Pelàez. "The Stored-Program Computer: Two Conceptions." *Social Studies of Science*, vol. 29(3) (1999), 377.
[80] Simon Lavington, *A History of Manchester Computers* (Manchester: NCC Publications, 1975), 9.
[81] Pelàez, "The Sstored-Program Computer: Two Conceptions," 377.
[82] Alan Turing, "Proposed Electronic Calculator," *Catalogue: Turing's Automatic Computing Engine* (n.d.), 14.
[83] Turing, "Proposed Electronic Calculator", 16.

This tenth problem was essentially ignored, and further proposals for the ACE focused on applications to aircraft and explosives.[84] While Turing's proposal was detailed and his vision clear, the NPL leadership was reticent. Turing quickly grew frustrated with the NPL bureaucracy.[85] He left for a sabbatical at Cambridge at the end of 1947, then was appointed reader in mathematics at the Victoria University of Manchester in September 1948.[86]

While in Cambridge, he reunited with a friend from Cambridge University, David Champernowne, to design a chess-playing program. At this point, "program" simply meant "set of instructions." Combining their surnames, they devised the *Turochamp*.[87] It operated by a simple principle: a function that would score a board position based on the amount of material on the board, the vulnerability of the king, the number of pieces under attack, and other important factors known to amateur chess players. The program then used von Neumann's minimax procedure for choosing the move that would lead to the best outcome, given the constraint that the opponent is trying to find their best board position too.

Turing then went one step further, and proposed that the machine could "profit from experience." The machine could alter the function it uses to assign position valuations, test different functions, and adopt the one that gives the best results. He then asks himself: if a computer that can learn ends up outwitting its programmer, who should get the credit for a success? He invites us to "compare this with the situation where a Defence minister gives orders for research to be done to find a counter to the bow and arrow. Should the inventor of the shield have the credit, or should the Defence Minister?" The connection to Blechley Park hardly needs pointing out: chess-master cryptoanalysis were given orders to find a counter to German codebreakers. They saw the power in machines to outwit the enemy. Turing now confronted the power of his program to outwit him.

---

[84] John Womersley, "Memo from Womersley: 'ACE' Machine Project," *Catalogue: Turing's Automatic Computing Engine* (n.d.).
[85] Ibid., 233
[86] Martin Campbell-Kelly, "Programming the Mark I: Early Programming Activity at the University of Manchester." *IEEE Annals of the History of Computing*, vol 2(2) (1980), 132.
[87] The entire description of the *Turochamp* program in this and the following paragraph comes from Turing in Bowden, ed., *Faster Than Thought*, 292-295.

Turing's chess program lay dormant for two years while Turing was working the design of the first commercial general-purpose computer, the Ferranti Mark I in Manchester.[88] In 1951, when the Mark I was up and running, Turing tried adapting the chess program he had written in 1948 into computer code, but failed.[89] The machine simply did not have the speed or space. A colleague, Dietrich Printz, managed to get the Mark I to solve mate-in-two problems, but this did not interest Turing.[90] Printz's program was a brute-force one: simply assess all possible moves, replies, and counter-replies, and pick the branch of the tree that leads to checkmate. It took 15 minutes to solve problems that humans could solve in seconds; precisely the opposite of what you would expect from a time-saving electronic brain.[91] But Turing still wanted to try out his Turochamp program. He recruited computer scientist Alick Glennie, and executed the program by hand, calculating board positions move-by move in a dull but historic match. Each move took half an hour of manual calculation, and Turing swiftly lost. He soon after described Turochamp's play as "a caricature" of human chess-playing.[92] Nevertheless, the program worked, and Turing was proud enough of it that when asked to submit a piece for *Faster Than Thought*, the first edited collection of writings on computers in the UK, Turing chose to write about chess.[93]

During this same period, Turing's Bletchley Park chess companion Donald Michie was back at Oxford, having completed his undergraduate in classics and now studying biology. [94] Starting in 1948, he began devising a portmanteau program of his own, *Machiavelli*, with fellow codebreaker Shaun Wylie. Turing visited Michie in Oxford in September 1948, and claimed that *Machiavelli* "suffers from the very serious disadvantage that it does not analyse more than one move ahead".[95] Later, in Manchester, Turing tried to program *Machiavelli* into the Mark I, with little success. The life of *Machiavelli* and *Turochamp* seem to end here, in 1952.

---

[88] Campbell-Kelly, "Programming the Mark I," 132.
[89] Gary Kasparov & Frederic Friedel, "Reconstructing Turing's 'Paper Machine'," 4
[90] Hodges, *Alan Turing: The Enigma*, 555.
[91] Turing in Bowden, ed. *Faster than Thought*, 295
[92] Ibid., 294
[93] Ibid.
[94] Donald Michie, "Curriculum Vitae", *Donald Michie Home Page*, http://www.aiai.ed.ac.uk/~dm/dmcv.html
[95] Turing in Hodges, 488

Though these programs were never successful, they helped advance an important if hubristic idea: chess, which takes humans years of intensive practice to master, could be automated by simple rules, and executed on a universal computer. For the Bletchley Park crew, programming chess was not that different from the preceding war—a strategic match to understand and defeat an equal opponent. Like the German Enigma, chess was a code to be cracked.

Chess, however, is uncrackable. According to a 1950 estimate by the American information theory pioneer Claude Shannon, there are 10^{120} possible chess games, a comically large number. To put this in perspective: if every person on earth right now played a chess game every second from the beginning of the universe until the present, they will have played 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001% of all possible games. For this reason, the first complex board game to be successfully programmed into a computer was one less associated with Oxford dons and more with school-age children, but whose story is no less impactful to the history of computing: draughts.

Chapter 2: A Big Jump

The First Draughts: Christopher Strachey

After Alan Turing left the National Physical Laboratory in 1948, construction began on a smaller, "Pilot" version of the ACE, as a proof-of-concept for the more ambitious universal engine.[96] The Pilot ACE first ran in May 1950, and while it was meant only as a prototype, it wound up being used for scientific calculations for nearly a decade.[97] But the machine's story did not stop there. Unbeknownst to its designers and programmers, the Pilot ACE was about to enter the annals of computing history at the hand of a most unlikely innovator: a flamboyant schoolteacher named Christopher Strachey.

Christopher Strachey was born in November 1916 into a prominent London family.[98] His father was a cryptographer with the Cypher School, and had a lifelong passion for puzzles, chess, and bridge; his mother was trained as a mathematician. Strachey inherited these traits, and was known to play three-dimensional noughts and crosses is his head.[99] He studied mathematics and physics at King's College, Cambridge then took a wartime job doing physics with Standard Telephones and Cables.[100] After the war, he fulfilled a lifelong dream of becoming a teacher, eventually serving as schoolmaster at Harrow School in London, where he taught maths.

In January 1950, Strachey's friend Mike Woodger, an engineer at the National Physical Laboratory, introduced him to the Pilot ACE project.[101] Strachey was smitten. He immediately began working on a draughts program in his spare time. The program became an obsession, as evidenced by his notes, which are sprawled on the back of Harrow School

---

[96] Carpenter, "Turing and ACE," 233.
[97] Ibid., 233
[98] Martin Campbell-Kelly, "Christopher Strachey, 1916-1975: A Biographical Note," *IEEE Annals of the History of Computing*, vol. 7 no. 1 (1985), 19
[99] Ibid., 20
[100] Ibid., 21
[101] Lawrence Mielniczuk, "Biographical History," *Catalogue of the papers of Christopher Strachey, 1930-1983,* 2016, http://www.bodley.ox.ac.uk/dept/scwmss/wmss/online/modern/strachey-c/strachey-c.html#introduction

mathematics exercise sheets.[102] These notes indicate that Strachey was working continuously on the problem from early 1950 to mid-1951.[103] But the process of actually turning his program—what we would now call an algorithm—into machine-readable and executable code was a painstaking one. The first attempt, in May 1951, failed because it exhausted the machine's memory stores.[104] The second attempt in July failed because of programming errors.[105] In another instance, the program failed because of a literal loose screw.[106]

At some point early in 1951, Strachey learned from Woodger of the Ferranti Mark I, the world's first commercial computer, which had been built in Manchester.[107] The Manchester computers were the first to feature a new form of memory called William tubes. Williams tubes used the presence or absence of electric charge to represent bits (i.e. 1s or 0s in binary notation), and small dots inside the tube would light up when electrically excited.[108] One could therefore see the contents of the computer's memory by examining patterns of dots on the Williams tube. The computer could also display images through creative manipulations the contents of its memory stores. The Ferranti Mark I also had a "hooter" function that sent a pulse through a loudspeaker.[109]

Turing had written a *Programmers Manual* for the Mark I, and Strachey requested a copy around March 1951.[110] Turing's *Manual* was notoriously opaque, yet Strachey vowed to understand it in full, because the Pilot ACE was simply too small and too slow to run his draughts program.[111] According to oral histories of the Manchester Computing Machine Laboratory, Strachey began making regular visits to the lab in October or November 1951. [112] Strachey's own notes indicate that he was attempting to run his program in early 1952.

---

[102] Christopher Strachey Papers, MS. Eng. misc. b. 258/C.20
[103] Ibid.
[104] B. Jack Copeland, *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle* (Oxford: Oxford University Press, 2005), 165.
[105] Christopher Strachey Papers, MS. Eng. misc. b. 258/C.30
[106] Ibid.
[107] Campbell-Kelly, "Christopher Strachey," 24.
[108] Campbell-Kelly, "Programming the Mark I," 134.
[109] Ibid., 136.
[110] Christopher Strachey Papers, MS. Eng. misc. b. 260/C.40.
[111] Christopher Strachey Papers, MS. Eng. misc. b. 260/C.40.
[112] Campbell-Kelly, "Programming the Mark 1," 131.

[113] As Martin Campbell-Kelly tells it, “The programme was about 20 pages long (over a thousand instructions) . . . After a couple of errors were fixed, the programme ran straight through and finished by playing “God Save the King” on the “hooter” (loudspeaker). On that day Strachey acquired a formidable reputation as a programmer that he never lost.[114]

Strachey took a commercial computing machine whose intended uses included fluid dynamics, munitions research, and atomic weapon construction, and turn it into a draughts player.[115] He had achieved what Turing was unable to achieve with chess.

Why Draughts?

The game of draughts has existed in some capacity for at least 4000 years, making it many thousands of years older than chess.[116] Its origins are in Egypt, and it was played widely in the ancient world; it is mentioned in Homer’s *Odyssey* and in Plato’s dialogues.[117] It evolved slowly over the centuries, beginning first as a fixture of courts, then a regular pastime in medieval France, before moving to coffee shops in the 17th century. Chess borrowed from draughts the idea of promoting a piece if it reaches the opponent’s side, and draughts began being played on chess board, hence its alternative name, “checkers”.[118] In the late 19th and early 20th century, as game-players began devoting more time to their craft, chess, with its myriad combinations and complex tactics, began to be seen as a more sophisticated and even profound. Draughts, meanwhile, lost its status as a gentlemen’s game, and became viewed as a pastime to be played among friends. Its basic rules were simple enough for children, but the game was complex enough to be interesting.

This is precisely why Christopher Strachey chose the game. As Strachey wrote in his contribution to the 1953 compendium *Faster Than Thought*, “the game of draughts

---

[113] Christopher Strachey Paper, MS. Eng. misc. b. 258/C.29.
[114] Campbell-Kelly, “Programming the Mark 1,” 133.
[115] Anthony Gandy, *The Early Computer Industry: Limitations of Scale and Scope* (Palgrave Macmillan UK, 2013), 134.
[116] Charles C. Walker, “Origin of Checkers of Draughts.” *Checkers Magazine*. June 1988. http://www.chesslab.com/rules/CheckerComments4.html.
[117] Ibid.
[118] Arie van der Stoep, “The History of Draughts,” *Draughts History* (n.d.), http://www.draughtshistory.nl/OpeningEngels.htm

occupies an intermediate position between the extremely complex games such as chess, and the relatively simple games such as nim or noughts-and-crosses for which a complete mathematical theory exists. This fact makes it a rather suitable subject for experiments in mechanical game-playing".[119] The relative simplicity of draughts therefore made it perfect for the computers of the early 1950s, whose memory and processing speed could not handle chess.

These computers were designed by national laboratories for missile calculations and complex physics. How did months end up being spent programming them for a board game? We can gain some insight through some of Christopher Strachey letters to Alan Turing. The two maintained a sympathetic correspondence throughout the 1950s. In May 1951, Turing gave a talk broadcast on the BBC called "Can Digital Computers Think?" He argued that "if any machine can appropriately be described as a brain, then any digital computer can be so described".[120]

Strachey wrote to Turing during the broadcast, sharing that their thoughts align "extraordinary well" on the subject.[121] Strachey writes that "the crux of the problem of learning is recognizing relationships and being able to use them . . . This was brought home to me in a very striking manner when I was investigating the behaviors of various possible types of game-playing machines".[122] He then describes playing a game called "Nim" with a friend. The game involves laying out matchsticks in three piles. Each player picks up as many matches as they want from one pile, taking turns, and the last person to pick up a match wins. After a few rounds of the game, Strachey's friend was able to intuit an optimal winning strategy, despite having never played before. Strachey concludes that "one of the most important features of thinking is the ability to spot new relationships when presented with unfamiliar material".[123] The letter then concludes with a lengthy update on the state of

---

[119] Strachey in Bowden, ed., *Faster Than Thought*, 298.
[120] Alan Turing, "Can Digital Computers Think", *The Turing Digital Archive,* 15 May 1951, http://www.turingarchive.org/browse.php/B/5.
[121] Letter from Christopher Strachey to Alan Turing, 15 May 1951, Christopher Strachey Paper, MS. Eng. misc. b. 258/C.22.
[122] Ibid.
[123] Ibid.

the draughts program, including details on how he is modifying the game to exploit the Mark 1's superior memory and speed.

For Strachey, then, game-playing did not represent a small subset of human intellect. Rather, the skills embodied in a game like Nim or draughts represented the very essence of cognition. If computers could play games like humans, then they could think like humans, and better serve humans in business, in science, or in war.

I REFUSE TO WASTE ANY MORE TIME: Strachey's Talkative Draughts Program

Strachey continued to refine his draughts program throughout 1952. Programming the Mark I to play draughts is a matter of giving numerical valuations to various positions and choosing moves which lead to high-valuation positions, much like Turing's chess program.[124] In principle, the program could pick a move by considering each possible move in a given position, then each reply to those moves, then each counter in turn, selecting the initial move which is most likely to improve the valuation at the end of the game (i.e. lead to a win).

But this is impossible for two reasons. First, memory. The Mark I could only store six moves—three per side—in its memory. Moreover, it could not simultaneously store all of the factors listed above, so it only considered the number of pieces on the board when valuating a position. Second, and more saliently, time. At each position, there are roughly ten legal moves, so each increase of the search depth increases the computation time by a factor of ten. The computer can evaluate 10 moves per second, so to consider a move, a reply, and a counter-reply would take:

$$10^3 \text{ moves} * 0.1 \text{ seconds/move} = 100 \text{ seconds} = 1.66 \text{ minutes}$$

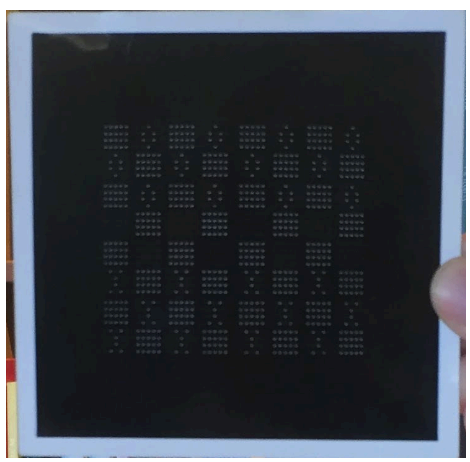
Exploring to one depth—or, to borrow language from Claude Shannon (which Strachey does not appear to be familiar with), one *ply*[125]—further would be 10^3 seconds, or 16

[124] Strachey in Bowden, ed., *Faster Than Thought*, 298.

[125] Shannon, "Programming a Computer for Playing Chess," 256.

minutes. A five-ply search would take 2.8 hours. This phenomenon is called the "combinatorial explosion." The number of combinations of options increases exponentially, exploding up to impracticality as more moves are considered. So in practice, the machine can decide how to move based only on looking ahead to a depth of 3 ply, i.e. a move, a response, and a counter. This allows the machine to play a "tolerable game," at least until it reaches the end-game.[126] This makes Strachey's draughts program the first instance of *heuistic programming*: a program that solves problems based on approximate strategies (like position valuations) rather than strict mathematical rules.[127]

Strachey exploited the Mark I's sophisticated Williams tube memory storage system to display the current of position on the board, and a preview of the next position while a player is deliberating (Fig 1).[128] After a number of input errors, the panel would display "IF YOU DON'T OBEY THE INSTRUCTIONS I CAN'T PLAY WITH YOU: TRY ONCE MORE." If this was followed by a final error, the user would see "I REFUSE TO WASTE ANY MORE TIME GO AND PLAY WITH A HUMAN BEING." These displays of "emotion" delighted Strachey.[129]



[126] "Draughts: General Considerations," Christopher Strachey Papers, MS. Eng. misc. b. 258/C.27
[127] David Link, "Programming ENTER," *Resurrection* vol. 60 (2012), 26.
[128] Campbell-Kelly, "Programming the Mark 1," 134.
[129] While finishing up draughts, he threw together a second program for the Mark I: an automatic love letter writer. Letters signed "M.U.C." (Manchester University Computer) began appearing in the computing lab's notice boards. Campbell-Kelly, "Christopher Strachey," 134.

Fig 1: A polaroid photograph of the Williams tube cathode-ray display used for Strachey's draughts program.[130]

In the process of writing his draughts game, Strachey proposed a number of changes that could be made to the design of both the Pilot ACE and the Manchester computers. "I have an idea about making the ACE do part of its own programming" he famously wrote to Woodger in May 1951.[131] Strachey wanted the computer execute the "very dull" process of turning a mathematically expressed algorithm (i.e. "if x, do y") into the equivalent of punched-card code. "It might be possible to make the machine really quite sly all on its own", he said.[132] Early computing was much more arduous; even short programs required dozens of punched cards to clearly specify the memory storage location of every quantity involved in a calculation.[133]

Making computers more programmer-friendly became the focus of the rest of Strachey's career. After the success of his draughts program on the Manchester machine, he was hired in November 1951 by the National Research Development Corporation as a consultant, where he remained for eight years. The NRDC was a postwar institution designed to take the scientific and technological and scientific developments from the public sector—mainly the research undertaken by the Defence Research Establishments during World War II—and transfer them to the private sector.[134] Computers were the obvious first step for patents and privatization, but the NRDC additionally worked on hover-craft and insecticides.[135] One of Strachey's first consulting projects was a new computer for Ferranti, the Pegasus. Strachey took extreme care to put the needs of the programmer first, and reduce the need for dull and extraneous operations. It was

[130] "Photographs of the monitor tube display during a game of draughts," Christopher Strachey Papers, MS. Eng. misc. b. 258/C.30.
[131] Letter from C. Strachey to M. Woodger, 25 May 1951, Christopher Strachey Papers, MS. Eng. misc. b. 258/C.23.
[132] Ibid.
[133] Christopher P. Burton, "Pegasus Personified – Simulation of a Historic Computer," *The Computer Conservation Society*, http://www.computerconservationsociety.org/Pegasus%20Personified-1.pdf
[134] Robert Budd, *Cold War, Hot Science: Applied Research in Britain's Defence Laboratories, 1945-1990* (Science Museum, 2002), 373.
[135] Jon Agar, *Science Policy Under Thatcher* (London: UCL Press, 2019), 67.

considered the first "user-friendly" computer. The Pegasus sold well[136] and was praised by its programmers and operators.[137] Its main practical application was for calculations by the Royal Airforce, and the Pegasus spent most of its lifetime being used to improve military aircraft.[138,139]

By mid-1952, Strachey's program was able to play a respectable game of draughts. He first presented his work to the computing community at the September meeting of the Association for Computing Machinery in Toronto. This conference was particularly important since the University of Toronto had purchased the first Ferranti Mark I outside of Manchester, and Strachey would be the representative Mark I expert.[140] His talk was humbly titled "Logical or Non-Mathematical Programmes," and it featured a detailed description of his draughts program.[141] The conference attendees were impressed by Strachey's work. He was hired as an official NRDC consultant on a project to use the Mark I to improve the flow of the St. Lawrence Seaway.

Within three years, Strachey had gone from maths schoolteacher to an international computing consultant, all on the strength of skills he gained programming draughts. A few months after his Toronto talk, Strachey was invited to contribute a chapter to the edited volume *Faster Than Thought*, which we have already seen was the first comprehensive book on computing written for a general audience. It contained a lengthy history of computing, detailed explanations of the Manchester machines, and a section on possible applications of computing, from theorem-proving to economics.[142] For a book on computing, which included descriptions of machines over which they achieved considerable mastery, Strachey wrote about draughts, and Turing about chess. This indicates that game-playing was viewed as an integral aspect of computing from its earliest

[136] Brian Parker, "Ferranti Pegassus, Perseus and Sirius", *Our Computer Heritage*, 5 March 2019, http://www.ourcomputerheritage.org/ccs-f3x1.pdf, 2.
[137] Donald Kershaw, "Experiences With Pegasus 1," *Computer Conservation Society*, 1994, http://www.computerconservationsociety.org/resurrection/res14.htm#f
[138] Burton, "Pegasus Personified", 4.
[139] Parker, "Ferranti Pegasus, Perseus and Sirius," 2.
[140] Cambpell-Kelly, "Christopher Strachey," 26.
[141] Christopher Strachey, "Logical or non-mathematical programmes" in J.W. Forrester and R.W. Hamming eds., "Proceedings of the 1952 ACM national meeting (Toronto)," September 1952.
[142] Bowden, *Faster Than Thought*.

days. Strachey's contribution was not merely a novelty. It was valued, earned him a prestigious job, and deemed worthy of appearing in his discipline's most important publication.

The most important outcome of the 1952 ACM conference was a symbolic one. On the other side of the Atlantic, unbeknownst to Strachey, there was someone working equally hard, on an equally ill-suited machine, on an equally path-breaking draughts program. Arthur Lee Samuel, an engineer at IBM, had been thinking about draughts (or as he would have called it, "checkers") for 5 years.[143] When he read the 1952 paper Strachey presented in Toronto, he was doubtless enraptured to see a full-fledged version of the program. He immediately returned to work, and within a few years had a program which far surpassed Strachey's and foreshadowed the next half-century of computer science research.[144] Strachey, on the other hand, didn't touch droughts again after the Toronto conference. Working on the problem for 2 years took a toll, and he later he summarized his experience building a draughts program as follows: "I had a look at it and came to the conclusion that it was not for me".[145]

Shades of Babbage: Arthur Samuel's Checkers Player

In many ways, Arthur Samuel's story mirrors Christopher Strachey's. Samuel, born in 1901, was trained as an electrical engineer at MIT, then spent a few years teaching before working at Bell Laboratories.[146] After World War II, he helped found a computer project at the University of Illinois, but left before the project was completed.[147] He arrived at IBM in New York in 1949, and began programming its first commercial computer to play checkers.

[143] I will refer to Samuel's program as "checkers" and Strachey's as "draughts" to avoid confusing the two.
[144] B. Jack Copeland, "What is Artificial Intelligence?" *Alan Turing Archiv* http://www.alanturing.net/turing_archive/pages/Reference%20Articles/what_is_AI/What%20is%20AI04.html
[145] Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973.
[146] Eric A. Weiss, "Eloge: Arthur Lee Samuel (1901-1990)," *IEEE Annals of the History of Computing*, vol. 14, no.3 (1992),55.
[147] Ibid., 57.

What began for Samuel as a simple pet-project turned into a major focus of his career. He was later championed as a pioneer of non-mathematical computing in America.[148]

Samuel's interest in checkers began in 1947, while helping design the first computer at the University of Illinois at Urbana-Champaign. The project was floundering and running out of funding. Someone suggested to Samuel that he should incorporate into the computer something showy to attract attention, and ultimately money.[149] One cannot help but think of Babbage doing the exact same with his Analytical Engine a century prior. It was at this time that Samuel learned about Claude Shannon's writings on chess. Shannon had been touring the country lecturing on the theory behind computer chess.[150] He was the first person to write a popular account of how a computer might be repurposed for chess, in an article for *Scientific American*.[151] At the time, in particular in the United States, the idea of using a computer to play chess seemed to many absurd—it was as if Shannon had suggested the game be played on a slide-rule or abacus.[152] Though Shannon's ideas were sound, this was 1950, and neither he nor anyone else was able to turn these principles into a workable chess program.

Samuel decided to try to write a checkers-playing game because, like Strachey, he recognized that checkers was the perfect middle-ground between simplicity and strategy. As he tells it: "I started writing a program for a machine that did not yet exist using a set of computer instructions that I dreamed up as they were needed".[153] Three years and a job switch to IBM later, Samuel's program worked for the first time. It took a further two years of tinkering before it could play an "interesting game".[154] Strachey himself saw the program in 1957, and wrote in his private notebooks that Samuel's machine "plays a very tolerable game".[155]

---

[148] John McCarthy & Edward A. Feigenbaum, "In Memoriam: Arthur Samuel, Pioneer in Machine Learning." *AI Magazine*, 11(3) (1990), 10.
[149] Schaeffer, *One Jump Ahead*, 87.
[150] Schaeffer, *One Jump Ahead*, 87.
[151] Claude Shannon, *A Chess-Playing Machine*, 48.
[152] Stephanie Dick, "AfterMath," 4.
[153] Schaeffer, *One Jump Ahead*, 87.
[154] One Jump Ahead, pg 88
[155] "America Trip", Christopher Strachey Papers, MS. Eng. misc. b. 297/H.2

Samuel was such a deft programmer that he was able to write programs faster than the team at IBM could improve its machines to a standard capable of playing them.[156] So for each incremental improvement in hardware, Samuel's checkers programs ended up being the stress test that the new components worked. As Pamela McCorduck describes it: "the machines stood in a row, playing ghostly games of checkers with their programmers in the hours between midnight and eight, being tested to go into the world and do accounting, inventory control, and other sober tasks by playing the game old men play with their grandchildren".[157] Once again, we see that game-playing is not a mere diversion. In the case of IBM in the 1950s, machines were deemed fit for sale on the basis of their ability to run a checkers program.

Despite the success of Samuel's checkers program, IBM was not keen to publish the results. The company was hesitant to talk about artificial intelligence publicly, since computers were meant to help humans, not usurp them.[158] It took until 1959 for the details of the program to be published in IBM's internal journal, and until 1963 for the write-up to appear in widely circulated press, as part of the book *Computers & Thought*.[159] In 1953, Samuel was shipped to Europe on a fact-finding mission: IBM was worried that the Europeans, and especially the British, were ahead of the US in computing.[160] Samuel was impressed by the frenzy of activity in the UK, and was able to use his checkers program as an entry ticket into laboratories that were otherwise unwelcoming.[161]

In February 1956, Samuel's program was featured in a televised promotional campaign for IBM. The company's stock rose by 15 points overnight, confirming his suspicion that a checkers program would attract funding for his computer projects. A few years later, prompted by interest from the artificial intelligence research community, Samuel decided to stage a game between his program and a human checkers champion. Samuel's program won against Robert Neely, a self-proclaimed master, and many

---

[156] McCorduck, *Machines Who Think*, 197.
[157] McCorduck, 197
[158] Weiss, "Arthur Lee Samuel," 63.
[159] Samuel in Feigenbaum & Feldman, ed., *Computers and Thought*, 71.
[160] Weiss, "Arthur Lee Samuel," 64.
[161] McCorduck, *Machines Who Think*,177.

publications celebrated the immense achievement of a computer playing at the highest levels of the game.[162] But this is not quite accurate. Nealey, a blind amateur checkers player, was ranked below the highest level "master" title according to every available source. His only formal title came in the unsurprisingly uncompetitive Connecticut state championship, four years after his match with Samuel's program.[163]

But at the time, these mitigating factors were not known, and Samuel was rightly celebrated for making a groundbreaking contribution to the budding field of AI.[164] His checkers program, and his overall work improving IBM's computing capabilities, earned him in late 1955 an invitation to a small summer workshop being organized at Dartmouth University. At the Dartmouth workshop, Samuel gave a talk on his checkers program.[165] Ray Solomonoff, a pioneer of machine learning, was amazed by the fact that Samuel's program could beat its programmer.[166] Turing's fears about credit attribution for programs that overtake their programmers was unfounded: Samuel got all the praise for his program's success. But at the Dartmouth workshop, Samuel made clear that he was not interested in checkers *pe se*; he was interested in checkers insofar as it can help us understand how humans learn and how to make computers learn.[167] I worry that Samuel might have been lying to himself and his audience with such a statement. Samuel staged publicity matches, appeared in televised stock-raising promos, worked on his checkers program for twenty years, and never systematically investigated how humans think about checkers, or anything else, for that matter—these are not the actions of someone who cares about about checkers for its applications to learning. We will see shortly that whether a machine could learn at all was a hotly contested question.

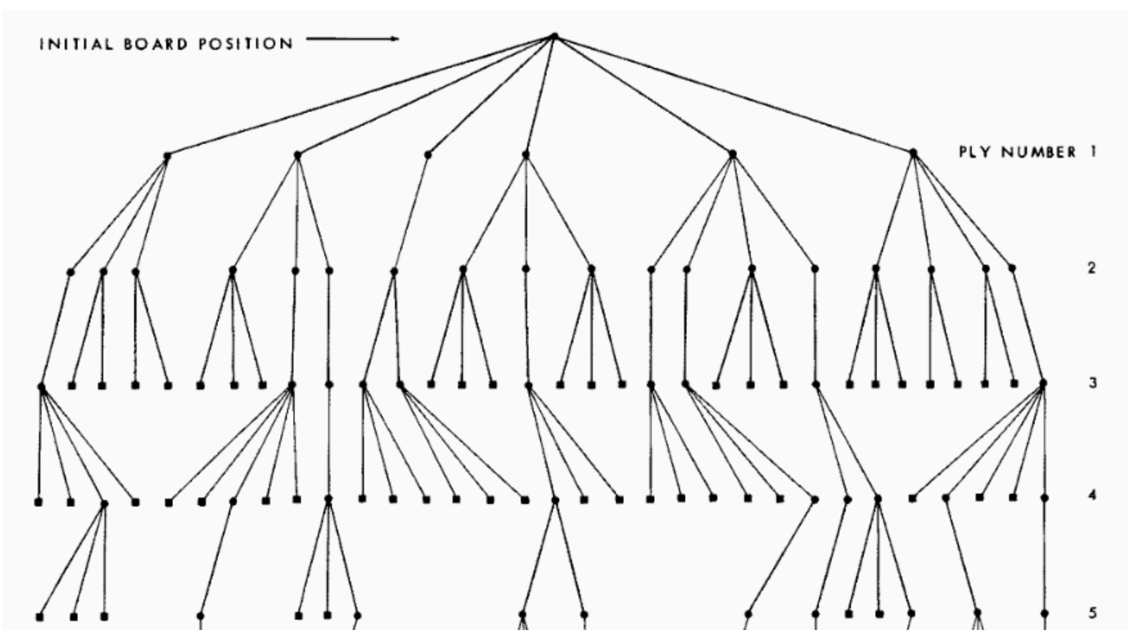
Machines Who Learn

---

[162] Schaeffer, *One Jump Ahead*, 90.
[163] Ibid., 91.
[164] Ibid., 91.
[165] Grace Solomonoff, "Ray Solomonoff and the Dartmouth Summer Research Project in Artificial Intelligence, 1956," *The Dartmouth AI Archives* (n.d.), http://raysolomonoff.com/dartmouth/dartray.pdf
[166] Ibid., 19.
[167] Ibid., 12.

Samuel's checkers program performed better than Christopher Strachey's for a number of reasons. For one, Samuel's valuation function took into account more than just the number of pieces on each side. It also included the number of kings and the proximity of pieces to being "kinged".[168] Because of the IBM machine's superior memory and processing speed, Samuel's program could look ahead as many as twenty ply.[169] And unlike Strachey's draughts, which always use a three ply look-ahead, Samuel's checkers program would vary the number of ply based on the game situation.

But as we have seen, with a greater search depth comes a greater cost in time. Picture any given board position as a dot, and each possible move from that position as a branch emanating from that dot (Figure 2). Each of those position will then have many counter-moves that emanate from it in turn. The standard minimax procedure begins from the bottom of the tree and works its way up, choosing the move that yields the best outcome for the program, given the opponent's goal to minimize the program's performance. Samuel needed to devise a way to take this unwieldly branching tree of possible moves—as many as ten unique moves per position per player—and remove branches.



[168] Samuel, "Some studies in machine learning", 213.
[169] Ibid., 214.

Figure 2: A simplified search. Samuel needed to determine how to avoid searching all possible branches of an unwieldly game tree.[170]

One solution to this problem was to save various positions in the computer's memory.[171] Samuel took advantage of the fact that there existed many annotated checkers games, with examples of good moves in various common positions. Samuel used one particular book called *Lee's Guide to Checkers*.[172] He placed these games in the computer's memory stores, enabling the computer to access these positions and quickly make human-like moves. Additionally, he had the computer store some of its own previous games, so that if it ended up in a position it had already encountered, and that position had led to a victory, it would repeat its strategy from before.

A later version of his program included an even more sophisticated mechanism to choose which branches were worth exploring. Given the minimax algorithm, some branches can be ruled out *a priori* because they contain positions which no rational opponent would select.[173] These branches do not need to be evaluated, saving time and memory. This algorithm was coined "alpha-beta pruning" by John McCarthy, and Samuel's program was the first to implement it, though it was invented multiple times independently in the 1950s.[174]

Finally, and most importantly, Samuel incorporated "learning" into his program, though many, including Christopher Strachey himself, would go on to dispute this terminology. He decided to let the computer play games against itself.[175] One version of the computer would stay the same in every game, but the other version of the computer-player would adjust its valuation function in response to the outcome of the previous game. In particular, the coefficients in front of the various terms in the evaluation

[170] Samuel, "Some studies in machine learning," 213.
[171] Samuel, "Some studies in machine learning," 215.
[172] McCarthy & Feigenbaum, "Arthur Samuel," 10.
[173] Samuel, "Some experiments in machine learning using the game of checkers. II—Recent progress," *IBM Journal* (1967), 603.
[174] Allen Newell & Herb Simon, "Computer science as empirical enquiry: symbols and search," *Communications of the ACM* vol. 19(3) (1976), 123.
[175] Samuel, "Some experiments in machine learning," 220.

polynomial—number of pieces, pieces under attack, proximity to kinging—would be free to vary. The program could therefore iterate and improve. Once the program improved sufficiently against its static opponent, Samuel would make the improved version the static one, and allow for another round of improvements.[176]

This, in the mid-1950s, was the invention of machine learning. Samuel's 1959 paper was the first to use that phrase, and his checkers program has been described as "the first functioning artificial intelligence program".[177] It predated the other programs typically awarded that distinction, such as Newell, Simon, and Shaw's Logical Theorist.[178]

Samuel's program was so effective and influential that there was virtually no progress on checkers until the work of Jonathan Schaeffer in the 1990s, because funding bodies had assumed that Samuel had fully solved the game.[179] Aside from being impressive, it demonstrates a broader point about checkers programming: after Strachey and Samuel, there was essentially no progress for thirty years. Samuel's program had reached its peak performance by the early 1960s, and though his later paper had many original insights, they did not affect the broader computing community.[180] This is a running theme in AI: the pioneers of the field delivered quick but incomplete results, and over-promised what was possible. This led to major setbacks in the field in the 1970s.

However, just as Strachey's droughts program led to advances in the theory of programming languages and the design of the Mark I, so Samuel's programs were pivotal to the early development of IBM computers and machine learning as a whole. Checkers, then, was more than just a game or toy example; it was where many standard concepts in computer science—hash tables, minimax, alpha-beta pruning, machine learning—originate.

Lessons Learned

[176] Samuel, Some experiments using the game of checkers, 1959
[177] Weiss, "Arthur Lee Samuel," 69.
[178] Crevier, *AI: The Tumultuous Search for Artificial Intelligence,* 45.
[179] Schaeffer, *One Jump Ahead*, 97.
[180] Schaeffer, *One Jump Ahead*, 97.

Why did Strachey dispute calling Samuel's program "learning"? Strachey had seen Samuel's program in 1957 on a visit to IBM, and this is perhaps the root of his skepticism. After hearing that Samuel designed a "learning machine," Strachey remarked in his notebook that "'learning' consists in recognizing positions already valued & using this value—this increases the depth of the valuation".[181] The quotation-marks around "learning" are important: Strachey always objected to the to the "miscellaneous and irresponsible use of words like 'learning', which have no very clear meaning. They are emotive terms. I do not believe that Samuel's checkers player is in any general sense a learning program".[182]

His reasoning can be gleaned from comments he made at the 1958 Mechanisation of Thought Processes conference at the NPL in Teddington, which was the most important AI conference to take place the UK at that time. Many of the leading AI pioneers from the United States, including Marvin Minsky, John McCarthy, and Warren McCulloch, made the trip. Of the 200 delegates, one-third came from overseas.[183] Strachey, commenting on a talk entitled "Learning Machines," notes that he believes that computer scientists have erred in referring to optimization programs as learning programs.[184] For Strachey, his draughts program and Samuel's checkers programs are merely an optimum-seeking machine: their only task is optimizing some definite quantity—the position valuation—which requires mitigating the combinatorial explosion and finding the best branches to search. Strachey was consistently clear: Samuel's program "is an optimizing program. I do not call optimizing programs learning programs".[185] Strachey was generally pessimistic about the Teddington conference as a whole. In a personal note, he wrote in reference to computer learning that "in spite of all the publicity which it attracted, the most striking thing which emerged from the Symposium on the Mechanisation of Thought Processes at Teddington was that there was no sign of a break through on this problem".[186]

---

[181] "America Trip", Private notebook, Christopher Strachey Papers, MS. Eng. misc. b. 297/H.2.
[182] Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973.
[183] NPL, *Mechanisation of Thought Processes*, ii.
[184] Ibid., 507.
[185] Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973.
[186] Symposium on the Mechanization of Thought Processes, Christopher Strachey Papers, MS. Eng. misc. b. 297/H.3.

Samuel's own view was that a computer should not learn like a human learns, in the same way that an airplane should not fly like a bird flies.[187] For Samuel, this meant using techniques—storing annotated games, and improving from self-play—that humans do not typically use, but which could nevertheless be profitable for a machine. Strachey would later complain that while Samuel's program was able to beat Samuel himself, it was never a better player than the people whose games Samuel fed into the machine.[188] On this point, however, Strachey misunderstood the potential of machine learning. Samuel's idea of self-play has proven to be one of the most important revelations in AI, and its modern reincarnation, reinforcement learning, powers the most powerful contemporary game-playing programs.[189] Regardless, we once again see that game-playing is hardly a frivolity. In early computing, ideas about what it meant for a computer to learn were inextricably tied to draughts.

More broadly, the Teddington conference made clear that AI research was deeply embedded in its Cold War context. A Russian delegate, Dr. A. P. Ershov, was scheduled to give a talk on automatic computing in the USSR, but ended up giving an impromptu talk on Soviet machine translation efforts.[190] Machine translation, especially between Russian and English text, was considered one of the most important tasks in early AI. Arthur Samuel's IBM 701 was first advertised as an "electronic brain" capable of executing translations that would serve "the national interest in defense or in peace".[191]

Some final biographical details about Samuel help situate his work within this milieu. Early in his career, Samuel began working part-time for the National Security Agency (NSA), and continued there even after he'd retired from IBM.[192] He was also the chairman of the Department of Defence Advisory Group on Electron Devices and

---

[187] McCorduck, *Machines Who Think*, 179.
[188] Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973.
[189] Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362(6419) (2018).
[190] NPL, *Mechanisation of Thought Processes*, iv.
[191] IBM, "701 Translator," *IBM Press Release*, 8 January 1954.
[192] Weiss, "Arthur Lee Samuel," 61.

coordinated IBM's role at the Offutt Air Force base in Nebraska.[193] Due to classification of military documents, it is not clear what precisely he was doing in any of these positions, though it is possible to speculate. For instance, Samuel likely served as a consultant on the IBM 7030 Stretch project, a computer built for atomic bomb calculations in Los Alamos, which was additionally shipped to the NSA.[194,195] Like many computer scientists of the era, Samuel's later research at Stanford was supported by a Department of Defense grant, provided by DARPA (Defense Advance Research Projects Agency).[196] Just as with Turing and Strachey before him, Samuel's work was taking place against the backdrop of military conflict. Military concerns, in the past and in the present, have shaped computer science research, and that research has in turn informed military applications. We now return to chess, whose history is even more intertwined in 20th century computing and geopolitics.

---

[193] Ibid., 61.
[194] Letter from Christopher Strachey to S.N. Alexander, Christopher Strachey Papers, MS. Eng. misc. b. 297/H.2.
[195] Computer History Museum, "Timeline of the IBM STRETCH/Harvest Era (1956-1961)," *Computer History Museum Archive* (n.d.), http://archive.computerhistory.org/resources/text/IBM/Stretch/102636400.txt
[196] Weiss, "Arthur Lee Samuel," 62.

Chapter 3: The Politics of Chess

The Nuclear Chess Match

As we have already seen, chess was one of the first games analysed using the new tools of game theory. But in addition to chess, there was a second two-player zero-sum game receiving a great deal of attention in the late 1940s: nuclear war. With an arms race with the Soviet Union underway, thought leaders in the United States began to conceptualize their position as analogous to chess.[197] In thermonuclear war one player's win is necessarily another player's loss, and each player must act with the knowledge that their opponent is trying to counteract its advances. The promise of game theory, as sold to the US government, was that it could rationally "solve" these war games using techniques like minimax.[198] In the late 1940s, researchers at the Institute for Advanced Study at Princeton, where von Neumann was based, and the RAND (Research And Development) Corporation began modelling common war scenarios—duels between fighter and bomber aircraft, or the allocation of scarce resources across battlefields were—as two-person zero-sum games. The central metaphor was to view America and the Soviet Union occupying "symmetric positions, adversaries playing the same pieces across a shared chessboard".[199]

Norbert Weiner, writing in 1961 in the second edition of his manifesto *Cybernetics*, reaffirms this strong connection between the mechanization of game-playing and the mechanization of war.

> What is true of games of physical encounter is also true of contests in which the intellectual element is stronger, such as war and the games which simulate war . . . This is true for classical war both on land and at sea, and is equally true with the new and yet untried war with atomic weapons. Some degree of

[197] Erickson et al., *How Reason Almost Lost its Mind*, 17.
[198] Ibid., 139.
[199] Ibid., 17.

> mechanization, parallel to the mechanization of checkers by learning machines, is possible in all of these.[200]

But while Weiner was happy to promote the war-game analogy, he also critiqued the minimax strategies being used in both. He wrote of minimax that "in war, which is a sort of game, this will in general lead to an indecisive action which will often be not much better than a defeat".[201] He then gives examples from history to show that understanding the eccentricities of your opponent is far more likely to lead to victory than "an attempt to play the perfect game against the perfect opponent".[202] "In a similar way," he adds "books on chess theory are not written from the von Neumann point of view. They are compendia of principles drawn from the practical experience of chess players playing against other chess players of high quality".[203] This, according to Weiner, is why chess programs have failed to attain human-level play: they assume their opponent is a machine too.

The connection between chess and the Cold War runs deeper than metaphors and models. Los Alamos, New Mexico was the site of the Manhattan Project, the top-secret construction site for the Little Boy and Fat Man bombs that were dropped on Japan in August 1945. After the war, Los Alamos quickly became a facility for building a weapon a thousand times more powerful, called a hydrogen bomb, a thermonuclear bomb, or the "super".[204] Many of the first computers, such as the ENIAC, which was originally designed for missile projections, were put to use doing calculations for the super.[205] Nicholas Metropolis, who was involved in the original Manhattan Project, returned to Los Alamos soon after the war to build a computer designed specifically for the lab, given the backronym MANIAC (Mathematical and Numerical Integrator and Computer).[206] The public reveal of the MANIAC took place at the September 1952 Association for Computing

---

[200] Weiner, *Cybernetics*, 2ed., 175.
[201] Ibid., 171
[202] Ibid., 171
[203] Ibid., 171
[204] Peter Galison and Barton Bernstein, "In Any Light: Scientists and the Decision to Build the Superbomb 1952-1954," *Historical Studies in the Physical and Biological Sciences,* vol. 19 no. 2 (1989), 268.
[205] Herbert L. Anderson, "Metropolis, Monte Carlo, and the MANIAC," *Los Alamos Science*, Fall 1986, 100.
[206] Ibid., 100.

Machinery conference in Toronto, the very same event where Strachey first presented his draughts program.[207]

The main programmer of the MANIAC was Stanislaw Ulam, a Polish mathematician who is the namesake of the modern thermonuclear weapon design.[208] In the mid-1950s, Ulam and his team began using the MANIAC to experiment with chess. They were aware of Turing and Shannon's work, but more saliently, they had heard of some recent Russian publications claiming a computer in Moscow had been coded to play chess.[209] One could speculate that, in a lab designed to compete with and "enclose" the Soviet Union, in an era in which a space race had just begun, the Los Alamos scientists might have felt the need to up their game.[210]

Because memory storage on mid 1950s computers was still scant, they designed a version of chess played on a 6x6 board, without bishops. The computer used a pure minimax search algorithm to a depth of 4 ply (two moves by each player), and each move took around twelve minutes to consider.[211] The program played a total of three games—one against itself, one against a master, and one against a woman who had learned the rules of the game a week prior—making it the first computer program to play a full game of chess, albeit one without bishops. Its authors believed that their program had legitimate scientific value. Watching the computer play could "illuminate the mechanism by which the human brain operates" so that "new insights can be gained into a significant area of knowledge: the organization of thought".[212] To this day "Los Alamos chess" is a popular variant of the game.

Two events, both outside the scope of this thesis, wrap-up this potted history of the relationship between chess and the Cold War. The first is the 1972 World Chess Championship between the American Bobby Fischer and the Russian Boris Spassky. For many, Fischer's victory was the beginning of the end of the Cold War, and represented

---

207 Howard B. Demuth, John B. Jackson, Edmund Klein, N. Metropolis, Walter Orvedahl, and James H. Richardson, "MANIAC." in "Proceedings of the 1952 ACM national meeting (Toronto)," September 1952.
208 Anderson, "Metropolis, Monte Carlo, and MANIAC," 99.
209 Kister et al., "Experiments in Chess," *Journal of the ACM* vol. 2(2) (1957), 174.
210 Edwards, *The Closed World*, 10
211 Kister et al., "Experiments in Chess," 175.
212 Kister et al., Experiments in Chess, 1777.

American supremacy in chess and in global affairs.[213] The second event is the defeat of Russian grandmaster Gary Kasparov by IBM's Deep Blue in a highly publicized 1997 match. [214] Deep Blue's victory represented machine supremacy, a half-century in the making.

One final curious connection between Los Alamos and the history of AI is that the first major conference on the history of computing was held at Los Alamos in 1976.[215] It was at this event that the details of Turing's wartime work at Bletchley Park was first revealed to the wider computing community. It is somewhat fitting that the reveal took place at Los Alamos, the United States' own secretive wartime facility. It also highlights the broader point that the history of codebreaking, artificial intelligence, nuclear weapons, and game-playing are deeply intertwined. That being said, we now return to a less bellicose setting—University College London—to explore the important contributions of Donald Michie.

Michie's Machines

After Bletchley Park, Donald Michie returned to Oxford, which is where he wrote his program *Machiavelli.* He was studying for a Master's degree in anatomy, followed by a DPhil in mammalian genetics in 1953.[216] He, too, was subject to secrecy laws, so was unable to share what he had done in the war. But within a few years, he became a leader in British computing, largely on the basis of his work in game-playing.

The second half of the 1950s was productive for computer chess in the United States. The Los Alamos group published their work in 1956. IBM engineer Alex Bernstein managed to program a the first full-board computer chess game one year later, to much fanfare. [217] Newell, Simon, and Shaw were making steady progress on heuristic

---

[213] Daniel Johnson, "Cold War Chess," *Prospect Magazine*, 19 June 2005, https://www.prospectmagazine.co.uk/magazine/coldwarchess
[214] McCorduck, *Machines Who Think*, xxix
[215] Michael R. Williams, "The First Public Discussion of the Secret Colossus Project," *IEEE Annals of the History of Computing* (2018), 84.
[216] Michie, "Curriculum Vitae"
[217] Chess-Playing Programs and the Problem of Complexity

programming, and were clearly quite swept up in the pace of progress, because they predicted that a computer would be the world chess champion by 1967.[218]

Things did not look quite as optimistic in the UK, partially because so much of the interesting work was happening across the pond. In a 1963 letter to Christopher Strachey, Michie says that he has "never seen any specs of any other chess machine. I don't think the rules of Bernstein's were ever published. John McCarthy has written a program which is said to perform creditably, but again unpublished".[219] While there were occasional publications, it seems that chess in the mid-1950s was being worked out outside the traditional scientific publishing structure, despite the fact that many leading scientists were working on it.

After the 1958 Teddington conference, it was especially clear that the Americans had the upper hand. Large research facilities and a deep-pocketed Department of Defense made Stanford, Dartmouth, MIT, Carnegie Melon, and IBM able to train researchers and generate new ideas at a speed the UK simply could not match.[220] In a 1961 speech, Christopher Strachey noted that "the whole subject of programming…is in a very much more advanced state in America than it is in this country".[221] Strachey blames the inadequacy of British computers, which had not advanced much since Ferranti's innovations of the early 1950s, which "more or less totally inhibited work on this subject in Universities".[222] Looking specifically at chess, we can see that the British landscape was "lagging, most pathetically, compared to the Americans" to use Michie's words.[223] In 1962, a journalist contacted Michie offering money to stage a competition against a computer program and his newspaper's chess correspondent.[224] It had already been four years since Bernstein's program did the

---

[218] Allen Newell & Herb Simon, "Heuristic Problem Solving: The Next Advance in Operations Research," *Operations Research*, vol 6 no. 1 (1958), 7.
[219] Letter from Donald Michie to Christopher Strachey, 1 Jan 1963, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.
[220] McCorduck, *Machines Who Think*, 131
[221] "Developments in mathematical and information processing languages," Christopher Strachey Papers, MS. Eng. misc. b. 297/H.6.
[222] Ibid.
[223] Letter from Donald Michie to Christopher Strachey, 22nd June, 1962, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.
[224] Letter from Donald Michie to Christopher Strachey, 1 Jan 1963, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.

same in the United States. In the US, undergraduate students at MIT were writing chess programs for their theses; [225] in the UK, no one had yet programmed a computer to play chess. Within a couple years, however, Michie would go on to almost singlehandedly reinvigorate artificial intelligence research in the UK.[226]

Michie was by this time based at the medical faculty of the University of Edinburgh. [227] He spent a majority of his free time on computing, and especially game-playing. For Michie, "games provide a microcosm of intellectual activity," and they clearly brought him more fulfillment than his medical work.[228] In 1961, in response to a challenge that machines could not learn, Michie invented a physical system out of matchboxes that could learn to play noughts and crosses.[229] He called it the Matchbox Educable Noughts And Crosses Engine, or MENACE.[230] The set-up works in such a way that if MENACE wins the game, it is "rewarded" in such a way that makes the moves that led to victory become more probable. Similarly, if the human wins, then the moves that led to MENACE's defeat are removed. In response to a dare, Donald Michie had invented reinforcement learning.[231]

Michie used the word "reinforcement" to describe his program, echoing the language from behaviorist psychology.[232] Soon after constructing the MENACE, he programmed it into the Ferranti Pegasus 2—a new version of the same Pegasus computer Christopher Strachey had designed—and ran some simulations.[233] This allowed him to adjust the reinforcement parameters, i.e. determine the number of beads to add which will lead to the fastest learning. Open any machine learning journal in 2019, and you will find these exact same types of experiments.

Community and Overconfidence

[225] Alan Kotok, "A Chess Playing Program for the IBM 7090 Computer," *Undergraduate Thesis*, June 1962.
[226] Margaret Boden, "Donald Michie (1923-2007)," *Nature*, vol 448 (2007), 765.
[227] Ibid., 765
[228] Donald Michie, "Experiments on the mechanization of game-learning Part I. Characterization of the model and its parameters," *The Computer Journal*, vol. 6(3) (1963),
[229] Ibid.
[230] Margaret Boden, "Donald Michie (1923-2007)," *Nature* vol 448 (2007), 765.
[231] Rich Sutton, "History of Reinforcement Learning" *Incomplete Ideas Blog*, 1 April 2005, http://www.incompleteideas.net/book/ebook/node12.html.
[232] Michie, "Experiments on the mechanization of game-learning," 232.
[233] Michie, "Experiments on the mechanization of game-learning," 234.

Michie's pioneering work was not going unnoticed. By 1962, he had job offers from Bell Telephone Laboratories, Labs, IBM, and a US Office of Naval Research-funded post Stanford.[234] He wound up travelling to Stanford to program a MENACE-like game on an IBM computer, but wrote that he "would infinitely prefer to develop this line in my own country, particularly since there is such a need to get things moving here".[235] The issue, as always, was funding. He eventually persuaded the Royal Society to provide him with "a few hundred pounds" to set up a small research group at the University of Edinburgh in 1963.[236] Within two years, he had established the Experimental Programming Unit, which later evolved into the Department of Machine Intelligence and Perception in 1966. It one of the first computer science departments in the UK (Manchester was accepting computer science undergraduates in 1964[237]), and the first dedicated principally to "machine intelligence".[238] It offered a graduate diploma whose courses included "heuristic problem-solving programs" and "game-playing and adaptive control programs".[239] Michie's own goals in the department were two-fold. The first was to build an intelligent robot. The second, of course, was to build a computer that could master chess.

Throughout the 1960s, Michie continued to build the artificial intelligence research community in the UK. In 1964 he established an "informal steering committee to consider starting a professional group for artificial intelligence studies", which included Christopher Strachey and his former Bletchley colleague I.J. Good.[240] At the same time, computing hardware was making large strides. The Ferranti Atlas 1, a state-of-the-art transistor-based supercomputer, went online in 1962. The Atlas Computer Laboratory in Oxfordshire housed

---

[234] Letter from Donald Michie to Christopher Strachey, 22nd June, 1962, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.
[235] Ibid.
[236] Boden, "Donald Michie," 765.
[237] Department of Computer Science, "History and Heritage," *University of Manchester,* https://www.cs.manchester.ac.uk/about/history-and-heritage/.
[238] Jim Howe, "Artificial Intelligence at Edinburgh Univeristy, A Perspective," *The University of Edinburgh School of Informatics,* June 2007, http://www.inf.ed.ac.uk/about/AIhistory.html.
[239] "Diploma in Machine Intelligence and Perception: University of Edinburgh," Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.
[240] Letter from Donald Michie to Christopher Strachey, 22 September 1964, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.

one of the devices and became a hub for computing research nation-wide.[241] By September 1964, Strachey commented to Michie: "I don't think we are behind the Americans — if anything the reverse".[242] Within two years of their initial pessimistic diagnoses, things were turning around. The high would be short-lived.

Michie had in this period struck up a fast friendship with John McCarthy.[243] John McCarthy had organized the Dartmouth Conference in 1956, and was in many ways the ringleader of the American AI research community.[244] McCarthy was perhaps the most avid chess programmer (though nowhere close to the best player) among the group. Michie organized an annual Machine Intelligence Workshop at the University of Edinburgh, and McCarthy was a perennial attendee, usually giving talks on game-playing.[245,246] At the 4th workshop, in the summer of 1968, Michie hosted a party, to which he invited 22-year-old English International Master (one level below the coveted "grandmaster" title) David Levy, who had just won the prestigious Scottish Chess Championship.[247] At this party, McCarthy claimed that a computer could beat Levy at chess within a decade.[248] They decided to bet on it, and Michie got in on the action as well. By 1971 there were four scientists pitted against Levy, for a sum of £1250. In 1974, Michie an additional side-bet, namely that Levy would lose to a program Michie himself had written. Levy unflinchingly accepted.

The Levy bet reveals the extreme hubris of the leading AI researchers in the late 1960s and early 1970s. In 1968, when the bet was made, a computer could barely beat an unskilled human. There had been few conceptual breakthroughs since minimax, alpha-beta pruning, and Bernstein's incorporation of book-knowledge. Computers were getting more

---

[241] Lavington, *A History of Manchester Computers*, 37.
[242] Letter from Christopher Strachey to Donald Michie, 25 September 1964, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.25.
[243] John McCarthy Papers, Box 6 Folder 7 is filled with humorous and warm letters between them.
[244] McCorduck, *Machines Who Think*, 130-131
[245] Bernard Melzer & Donald Michie, ed., *Machine Intelligence, Volume 4* (Edinburgh: University of Edinburgh Press, 1969).
[246] John McCarthy sc0524_1995-247_b27_f02.pdf
[247] Chess Scotland, "Scottish Championship 1968," *Scottish Chess,* https://www.chessscotland.com/documents/history/1968champ.htm
[248] Computer History Museum, "Oral History of David Levy,", 8 September 2005, https://www.computerhistory.org/chess/orl-4345632d88ad1/.

powerful every year, but this simply meant that computers could play a bit faster, not better. As we will soon see, this foolhardy overoptimism could not last indefinitely.

Chapter 4: The Lighthill Report

Unrest in Edinburgh

While Donald Michie was making high-profile bets with chess masters, he was also fighting fires at the Department of Machine Intelligence and Perception in Edinburgh.[249] Michie and his two departmental co-founders Richard Gregory and Christopher Longuet-Higgings, one a psychologist, the other a cognitive scientist, disagreed considerably about the aims of their research program.[250] As his betting habits suggest, Michie was known to overpromise and that was certainly the case for the Edinburgh robot, FREDDY.[251] Michie was interested primarily in "heuristic search and game-playing" to apply them to the construction of a general-purpose robot. [252] In his papers leading up to the construction of FREDDY, Michie does not mention the work of Gregory and Longuet-Higgins on human cognition.[253] In fact, Michie seems to have little interest in actual human cognition, and in most of his writings speaks in extended analogies between chess and robot spatial navigation.[254]

In 1970, only 3 years into the existence of the department, Richard Gregory frustratedly resigned and took up a post at Bristol University.[255] This caused intense discord within the department. Multiple of the research groups were renamed, restructured, or removed entirely. The Department leadership was forced to step aside in favour of an independent steering committee. Before long, the Science Research Council, which funded the research in Edinburgh, caught wind of the turmoil.[256]

[249] Howe, "Artificial Intelligence at Edinburgh University"
[250] Ibid.
[251] Boden, "Donald Michie," 765
[252] Letter from Donald Michie to Christopher Strachey, 9 September 1969, MS. Eng. misc. b. 300/J.25.
[253] Donald Michie, "Machines and the Theory of Intelligence," *Nature* vol 241 (1973), 507.
[254] Donald Michie, "Programmer's Gambit," *New Scientist*, 17 August 1972.
[255] Howe, "Artificial Intelligence at Edinburgh University"
[256] Ibid.

Lighthill Weighs In

Prompted by the situation in Edinburgh, the head of the Science Research Council, Brian Flowers, sought an independent assessment of the state of artificial intelligence in the UK. He commissioned Professor Sir James Lighthill, at the time holding the prestigious chair of Lucasian Professor at Cambridge—a post previously held by Isaac Newton and Charles Babbage. Lighthill surveyed dozens of British and American researchers in the field, asking for their comments on AI's "promise in the short term and in the long term future".[257] Flowers and the Science Research Council found that "some of the proposals in the field [were] getting very big",[258] meriting the whole field for review.

One of the scientists Lighthill contacted was Christopher Strachey, who had by the 1970s turned decisively away from the "artificial intelligentsia," as they came to be pejoratively known.[259] In particular, Lighthill wrote to Strachey to ask for some comments, seeking the perspective of someone in an adjacent field. Strachey's reply is instructive. He says that AI "has suffered more from the incautious comments of its friends and the trivializing publicity it gets from the press than it has from any direct attack".[260] Strachey believed that problems in AI remained exceedingly difficult because researchers tended to "overestimate the speed and power of a computer, or possibly underestimate the combinatorial explosion which takes place if you try to use brute force methods".[261] This combinatorial explosion would go on to become a central pillar of Lighthill's critique of AI.

Lighthill submitted his report in March 1972 for publication the following year. Lighthill's general critique is that AI is attempting to be a bridge between two disciplines that exist quite happily separately. There is A: advanced automation, and C: central nervous

---

[257] Letter from Lighthill to Strachey, 7 February 1972, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.23
[258] Ibid.
[259] Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973.
[260] Strachey to Lighthill, 13 February 1972, Christopher Strachey Papers, MS. Eng. misc. b. 300/J.23
[261] Ibid.

system (i.e. neurophysiology). Artificial intelligence is trying to serve as a B: bridge (or "building robots") between the two.[262]

Chess featured prominently in Lighthill's criticisms of the field. He notes that "it is interesting to consider the results of all this work some twenty-five years after the researches aimed at chess-playing programs began: unfortunately these results are discouraging. The best programs play chess of only experienced amateur standard characteristic of county club players in England. Chess masters beat them easily".[263] In many ways, he was correct. Chess programs were not significantly stronger in 1973 than they were in 1963, and Lighthill had correctly identified a stagnation which the scientists themselves had failed to pick up on. That being said, Lighthill likely would not have found it plausible that within an additional twenty years of his report, a chess player would beat Gary Kasparov.

In June of 1973 a public debate was held as part of the television programme Controversy, featuring James Lighthill as the principal speaker. The panelists who were set to face off against Lighthill were Donald Michie, John McCarthy, and Michie's begrudged Edinburgh department-made Richard Gregory. This debate is a fascinating glimpse into the world of AI and its critics, among them in the audience Christopher Strachey. Lighthill's report led to dramatic funding cuts and prompted the first "AI winter," a period of decreased research activity in artificial intelligence. The state of AI research in the UK did not recover to its pre-Lighthill levels until Japan announced its Fifth Generation Computing project, which led to massive re-investments in AI.[264]

---

[262] Sir James Lighthill, "Artificial Intellingece: A General Survey," *SRC*, 1973.
[263] Sir James Lighthill, "Artificial Intellingece: A General Survey," *SRC*, 1973.
[264] Steve Woolgar, "Why Not a Sociology of Machines, The case of artificial intelligence," *Sociology,* vol 19, no.4, 557.

Conclusion: The Drosophila of AI?

In 1978, one decade after the initial bet was made, David Levy played a five-game match against the world's leading chess-playing computer, CHESS 4.7, at the Canadian National Exhibition in Toronto.[265] After the first game ended in a draw, Levy won two in a row, followed by a victory for CHESS 4.7, and Levy winning the fifth and final game. This 3 ½ – 1 ½ victory was a decisive win for humans, and blow to the egos and pocketbooks of Donald Michie and John McCarthy. What had gone wrong? After such rapid progress in the early 1950s, with draughts and chess pushing computers to new limits, why did things crash so swiftly?

One answer is that the field relied too strongly on chess as a model for artificial, and human, intelligence. Chess has been called the "drosophila" of AI, a reference to the common fruit fly, the model organism studied in genetics and developmental biology.[266] John McCarthy diagnoses AI's focus on chess by showing the inadequacy of that analogy: "computer chess has developed much as genetics might have been if the geneticists had concentrated their efforts starting in 1910 on breeding racing Drosophila. We would have some science, but mainly we would have very fast fruit flies".[267]

If the goal of studying game-playing was to understand the human mind, as it was for Newell, Simon, and Shaw, as well as Arthur Samuel and Alan Turing, then AI certainly failed in that respect. We now have neural network and reinforcement learning-based AI that can beat the best human players a hundred times over. Though these programs have been useful for chess education and have expanded the repertoires of grandmaster players, they have taught us very little about the workings of the human mind.

But this does not mean that game-playing was somehow a frivolity or waste of time. To the contrary: I have argued that game-playing has played a foundational role in the history of artificial intelligence and computing more broadly. Games were one of the first

---

[265] Nilsen, *The Quest for Artificial Intelligence,* 403.
[266] Nathan Ensmenger, "Is chess the drosophila of artificial intelligence? A social history of an algorithm." *Social Studies of Science*, vol 42 no. 1, 5.
[267] John McCarthy, "AI as sport." *Science* 276(5318) (1997): 1518.

uses for early computers like the Ferranti Mark I and IBM's 701. In the latter case, the game of checkers helped verify the hardware, and in both cases, non-numerical programming like draughts or chess led to dramatic improvements in programming languages. Games like chess and draughts also raised fundamental questions: what would it mean for a computer to learn? Who is responsible for a program that outwits its creator? These questions are still asked today.

At the same time, games like chess and draughts were no different from the other applications computers were put to during the Cold War. Chess served as the prototypical metaphor for an adversarial symmetric contest. All of the individuals highlighted in this thesis—Christopher Strachey, Arthur Samuel, Alan Turing, Donald Michie—were game-playing enthusiasts, but were also men of their time, and participated in the military structures that enabled their profession to exist and encouraged their work.

Game-playing, then, was a bridge, to borrow a term from James Lighthill. It tied together Bletchley Park, Los Alamos, and the UK's first research department in Edinburgh. It was an intellectual challenge, a test-case for further work, and an indispensable part of early computing.

Bibliography

Primary Sources

Archival Sources

Catalogue of the papers and correspondence of Christopher Strachey, 1930-1983. Bodleian Library, University of Oxford

MS. Eng. misc. b. 258 – Automatic Computing Engine
MS. Eng. misc. b. 297 – Symposium on the Mechanisation of Thought Processes
MS. Eng. misc. b. 300/J.23 – Lighthill
MS. Eng. misc b. 300/J.25 – Michie

John McCarthy Papers (SC0524) Department of Special Collections and University Archives, Stanford University Libraries, Stanford, Calif.

ARCH-1995-247 Box 2, Folder 27
ARCH-1995-247 Box 4, Folder 15
ARCH-1995-247 Box 6, Folder 7
ARCH-2012-055 Box 34, Folder 23
ARCH-2012-055 Box 38, Folder 25 & 26

The Royal Institution, London

NCUACS 100.5.01/C.1231, C.1232

Print Sources

Babbage, Charles. *Passages from the Life of a Philosopher*. London: Longman, Green, Longman, Roberts, & Green, 465.

Bernstein, Alex & Michael de V. Roberts, "Computer v. Chess-Player", *Scientific American*, vol. 198 no. 6, June 1958

Bowden, B.V., ed. *Faster Than Thought: A Symposium on Digital Computing Machines*. London: Sir Isaac Pitman & Sons Ltd., 1953.

Demuth, Howard B., John B. Jackson, Edmund Klein, N. Metropolis, Walter Orvedahl, and James H. Richardson, "MANIAC." in "Proceedings of the 1952 ACM national meeting (Toronto)," September 1952.

Education Testing Service, program vol. 1, No. 4, "Horizons of Science: Thinking Machines," 1958.

Feigenbaum, Edward A., and Julian. Feldman, ed. *Computers and Thought*. New York; London: McGraw-Hill, 1963.

Gandy, Anthony. *The Early Computer Industry: Limitations of Scale and Scope.* Palgrave Macmillan UK, 2013.

Golombek, Harry and Bill Hartston. *The Best Games of C.H.O'D Alexander.* London: Oxford University Press, 1976.

IBM, "701 Translator," *IBM Press Release*, 8 January 1954.

Kister et al. "Experiments in Chess," *Journal of the ACM*, vol 2(2) (1957):174-177.

Kotok, Alan. "A Chess Playing Program for the IBM 7090 Computer," *Undergraduate Thesis*, June 1962.

Lighthill, James. "Artificial Intellingece: A General Survey," *SRC*, 1973.

Maynard Smith, John & Donald Michie. "Machines That Play Games," *The New Scientist* no. 260, 9 November 1961

McCarthy, John, Marvin Minsky, Nathaniel Rochester, and Claude Shannon. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence: August 31, 1955." *AI Magazine* 27, no. 4 (2006): 12-14.

Melzer, Bernard & Donald Michie, ed., *Machine Intelligence, Volume 4* (Edinburgh: University of Edinburgh Press, 1969).

Michie, Donald. "Programmer's Gambit," *New Scientist*, 17 August 1972

Michie, Donald. "Machines and the Theory of Intelligence," *Nature*, vol. 241 (1973): 507-512.

National Physical Laboratory. *Mechanisation of Thought Processes: Proceedings of a Symposium Held at the National Physical Laboratory on 24th - 27th November 1958*. No. 10. London: H.M.S.O, 1959.

Newell, Allen, J.C. Shaw, & Herbert A. Simon, "Chess-Playing Programs and the Problem of Complexity," *IBM Journal of Research and Development*, vol 2(4) (1958):320-335.

Newell, Allen & Herb Simon, "Heuristic Problem Solving: The Next Advance in Operations Research," *Operations Research*, vol 6 no. 1 (1958):1-10.

Newell, Allen & Herb Simon, "Computer science as empirical enquiry: symbols and search," *Communications of the ACM* vol. 19(3) (1976):113-126.

Samuel, Arthur. "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol 3(3) (1959): 210-229.

Samuel, "Some experiments in machine learning using the game of checkers. II—Recent progress," *IBM Journal* (1967), 603.

Shannon, Claude "Programming a Computer for Playing Chess," *The London, Ediburgh, and Dublin Philosophical Magazine and Journal of Science*, 41 no. 304 (1950): 256-275.

Silver et al., "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362(6419) (2018):1140-1114.

Strachey, Christopher. "Logical or non-mathematical programmes" in J.W. Forrester and R.W. Hamming eds., "Proceedings of the 1952 ACM national meeting (Toronto)," September 1952.

Wiener, Norbert. *Cybernetics: Or Control and Communication in the Animal and the Machine*, 2ed. (Cambridge: MIT Press, 1961).

Web Sources

Chess Scotland, "Scottish Championship 1968," *Scottish Chess,* https://www.chessscotland.com/documents/history/1968champ.htm

Christopher P. Burton, "Pegasus Personified – Simulation of a Historic Computer," *The Computer Conservation Society*, http://www.computerconservationsociety.org/Pegasus%20Personified-1.pdf.

Computer History Museum, "Oral History of David Levy,", 8 September 2005, https://www.computerhistory.org/chess/orl-4345632d88ad1/

Controversy, "The General Purpose Robot is a Mirage," *BBC*, June 1973. http://www.chilton-computing.org.uk/inf/literature/reports/lighthill_report/overview.htm

Department of Computer Science, "History and Heritage," *University of Manchester,* https://www.cs.manchester.ac.uk/about/history-and-heritage/.

Freidel, Frederic and Garry Kasparov. "Reconstructing Turing's 'Paper Machine.'" *Chessbase*. 14 September 2017. https://en.chessbase.com/post/reconstructing-turing-s-paper-machine

Jack Copeland, "Catalogue: Turing's Automatic Computing Engine," 2019, http://www.alanturing.net/turing_archive/archive/index/aceindex.html

Kershaw, Donald. "Experiences With Pegasus 1," *Computer Conservation Society*, 1994, http://www.computerconservationsociety.org/resurrection/res14.htm#f

Daniel Johnson, "Cold War Chess," *Prospect Magazine*, 19 June 2005, https://www.prospectmagazine.co.uk/magazine/coldwarchess.

Michie, Donald. "Curriculum Vitae." *Donald Michie Home Page*. http://www.aiai.ed.ac.uk/~dm/dmcv.html

Parker, Brian. "Ferranti Pegassus, Perseus and Sirius", *Our Computer Heritage*, 5 March 2019, http://www.ourcomputerheritage.org/ccs-f3x1.pdf.

Solomonof, Ray. "Who plus Aug 21-25," 1956, The Dartmouth AI Archives, Box B, http://raysolomonoff.com/dartmouth/boxbdart/dart56ray812825who.pdf.

Turing, Alan. "Proposed Electronic Calculator," *Catalogue: Turing's Automatic Computing Engine* (n.d.), http://www.alanturing.net/turing_archive/archive/p/p01/p01.php

Turing, Alan. "Can Digital Computers Think", *The Turing Digital Archive,* 15 May 1951, http://www.turingarchive.org/browse.php/B/5.

Wall, William D., "Chess World War II Codebteakers," 2018, http://billwall.phpwebhosting.com/articles/chess_codebreakers.htm

Womersley, John. "Memo from Womersley: 'ACE' Machine Project," *Catalogue: Turing's Automatic Computing Engine* (n.d.), http://www.alanturing.net/turing_archive/archive/l/l46/l46.php

Secondary Sources

Agar, Jon. *Science Policy Under Thatcher*. London: UCL Press, 2019.

Agar, Jon. "What is science for? The Lighthill report and the purpose of artificial intelligence research," Unpublished.

Anderson, Herbert L. "Metropolis, Monte Carlo, and the MANIAC," *Los Alamos Science*, Fall 1986, 100.

Aspray, William. *John Von Neumann and the Origins of Modern Computing.* Cambridge: MIT Press, 1990.

Bloor, David. *Knowledge and Social Imagery.* Chicago: University of Chicago Press, 1976.

Boden, Margaret. “Donald Michie (1923-2007),” *Nature* vol 448 (2007), 765.

Bowker, Geoff. “How to be Universal: Some Cybernetic Strategies, 1943-70.” *Social Studies of Science*., vol 23 (1993): 107-27.

Budd, Robert. *Cold War, Hot Science: Applied Research in Britain's Defence Laboratories, 1945-1990*. Science Museum, 2002.

Campbell-Kelly, Martin. “Programming the Mark I: Early Programming Activity at the University of Manchester.” *IEEE Annals of the History of Computing*, vol 2(2) (1980):120-168

Campbell-Kelly, Martin. ‘‘Programming the Pilot ACE: Early Programming at the National Physical Laboratory,’’ *Annals of the History of Computing*, vol 3 (1981): 133–162.

Campbell-Kelly, Martin “Christopher Strachey, 1916-1975: A Biographical Note,” *IEEE Annals of the History of Computing*, vol. 7 no. 1 (1985): 19-42.

Collins, Harry. *Artificial Experts.* Cambridge: MIT Press, 1990.

Computer History Museum, “Timeline of the IBM STRETCH/Harvest Era (1956-1961),” *Computer History Museum Archive* (n.d.), http://archive.computerhistory.org/resources/text/IBM/Stretch/102636400.txt

Copeland, B. Jack. “What is Artificial Intelligence?” *AlanTuring.net.* May 2000. http://www.alanturing.net/turing_archive/pages/Reference%20Articles/What%20is%20AI.html.

Copeland, B. Jack. *Alan Turing's Automatic Computing Engine: The Master Codebreaker's Struggle.* Oxford: Oxford University Press, 2005.

Copeland, B. Jack. *Colossus: The Secrets of Bletchley Park’s Codebreaking Computers*. Oxford: Oxford University Press, 2006.

Dick, Stephanie K., “After Math: (Re)configuring Minds, Proof, and Computing in the Postwar United States” PhD dissertation, Harvard University, Graduate School of Arts & Sciences, 2014.

Dick, Stephanie K. "Of Models and Machines: Implementing Bounded Rationality." *Isis* 106, no. 3 (2015): 623-34.

Edwards, Paul. *The Closed World: Computers and the Politics of Discourse in Cold War America*. Cambridge: MIT Press, 1996.

Erickson, Paul, Judy L. Klein, Lorraine Daston, Rebecca Lemov, Thomas Sturm, Michael D. Gordin. *How Reason Almost Lost its Mind: The Strange Career of Cold War Rationality.* Chicago: University of Chicago Press, 2013.

Ensmenger, Nathan. "Is chess the drosophila of artificial intelligence? A social history of an algorithm." *Social Studies of Science*, vol 42 no. 1 (2011):5-30.

Heath, David and Derek Allum. "The Historical Development of Computer Chess and its Impact on Artificial Intelligence." *AAAI Technical Report WS-97-04.* 1997.

Hern, Alex. "2016: The year AI came of age." *The Guardian*. 28 December 2016.

Hodges, Andrew. *Alan Turing: The Enigma*, 2ed. New Jersey: Princeton University Press, 2014.

Howe, Jim. "Artificial Intelligence at Edinburgh Univeristy, A Perspective," *The University of Edinburgh School of Informatics,* June 2007, http://www.inf.ed.ac.uk/about/AIhistory.html.

Galison, Peter and Barton Bernstein, "In Any Light: Scientists and the Decision to Build the Superbomb 1952-1954," *Historical Studies in the Physical and Biological Sciences,* vol. 19 no. 2 (1989), 268.

Kline, R. R. "Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence." *IEEE Annals of the History of Computing* 33, no. 4 (2011): 5-16.

Latour, Bruno and Steve Woolgar. *Laboratory Life: The Social Construction of Scientific Facts.* Beverly Hills: Sage,1979.

Lavington, Simon. *A History of Manchester Computers,* Manchester: NCC Publications, 1975.

Lefler, Mar. *Chess Programming Wiki*, 2019, https://www.chessprogramming.org/Main_Page

Link, David. "Programming 'ENTER,'" *Resurrection* vol. 60 (2012): 23-31.

McCarthy, John & Edward A. Feigenbaum, "In Memoriam: Arthur Samuel, Pioneer in Machine Learning." *AI Magazine*, 11(3) (1990), 10.

McCorduck, Pamela and Ed Feigenbaum. *The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World* (Reading: Addison-Wesley, 1983).

McCorduck, Pamela. *Machines Who Think: A Personal Journey into the History and Prospects of Artificial Intelligence*, 2ed. Massachusetts: AK Peters, Ltd, 2004.

Mielniczuk, Lawrence. "Biographical History," *Catalogue of the papers of Christopher Strachey, 1930-1983,* 2016, http://www.bodley.ox.ac.uk/dept/scwmss/wmss/online/modern/strachey-c/strachey-c.html#introduction

Muggleton, Stephen. "Donald Michie," *The Guardian,* 10 July 2007, https://www.theguardian.com/science/2007/jul/10/uk.obituaries1

Olazaran, Mikel. "A Sociological Study of the Official History of the Perceptrons Controversy." *Social Studies of Science*, vol 26 (1996).

Pelàez, Eloína. "The Stored-Program Computer: Two Conceptions." *Social Studies of Science*, vol. 29(3) (1999): 359-89.

Penn, Jonnie. "AI thinks like a corporation—and that's worrying." *The Economist*, November 2018.

Plasek, Aaron. "On The Cruelty of Really Writing a History of Machine Learning." *IEEE Annals of the History of Computing*, vol 38(4) (2016).

Schaeffer, Jonathan. *One Jump Ahead: Computer Perfection at Checkers,* 2ed. New York: Springer, 2009.

Simon, Herbert A. & Jonathan Schaeffer. "Chapter 1: The Game of Chess." *Handbook of Game Theory with Economic Applications* 1 (1992): 1-17.

Solomonoff, Grace. "Ray Solomonoff and the Dartmouth Summer Research Project in Artificial Intelligence, 1956," *The Dartmouth AI Archives* (n.d.), http://raysolomonoff.com/dartmouth/dartray.pdf

Standage, Tom. *The Turk: The Life and Times of a Famous Eighteenth-Century Chess-Playing Machine.* Berkeley: Berkeley Books, 2003.

Sutton, Rich. "History of Reinforcement Learning," *Incomplete Ideas Blog*, 1 April 2005, http://www.incompleteideas.net/book/ebook/node12.html.

The Telegraph, "Professor Donald Michie," *The Telegraph*, 9 July 2017, https://www.telegraph.co.uk/news/obituaries/1556846/Professor-Donald-Michie.html

van der Stoep, Arie. "The History of Draughts," *Draughts History* (n.d.), Archived by the Wayback Machine at https://web.archive.org/web/20121029041906/http://www.draughtshistory.nl/OpeningEngels.htm

Walker, Charles C. "Origin of Checkers of Draughts." *Checkers Magazine*. June 1988. http://www.chesslab.com/rules/CheckerComments4.html.

Weiss, Eric A. "Eloge: Arthur Lee Samuel (1901-1990)," *IEEE Annals of the History of Computing*, vol. 14, no.3 (1992):55-69.

William Clark, Jan Golinski, and Simon Schaffer. *The Sciences in Enlightened Europe*. (Chicago: University of Chicago Press, 1999), 158-160.

Williams, Michael R. "The First Public Discussion of the Secret Colossus Project," *IEEE Annals of the History of Computing* (2018):84-87.