Alan Turing – een enigmatisch genie

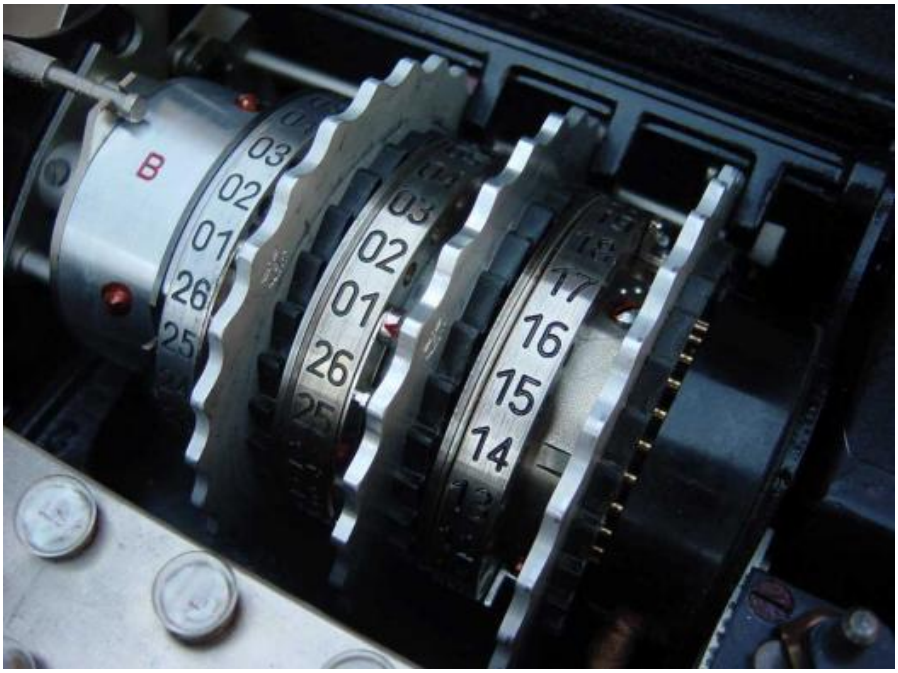
– door Herbert Tulleken –

Alan Turing (1912-1954), die vooral bekend is geworden als kraker van de Duitse Enigma encryptiemachine gedurende de Tweede Wereldoorlog, was in velerlei opzicht zelf een enigma (Grieks voor raadsel). In het navolgende wordt vooral Turing's leven vanaf 1939 besproken, startend in Bletchley Park, het hoofdkwartier van de Britse geheime dienst MI6, ten noorden van Londen.

Hier slechts een paar woorden over de jonge Alan: eenzaam, stotteraar, excentriek, te slim voor zijn leeftijd (wizz kid), gepest op kostschool (sissy boy) en als grootste hobby het oplossen van lastige puzzles (cryptoraadsels, schaakproblemen, etc). Jongvolwassen reeds een wiskundig genie, met vele belangrijke publicaties op zijn naam op het gebied van onder meer groepentheorie en Riemann's Zeta-functie.

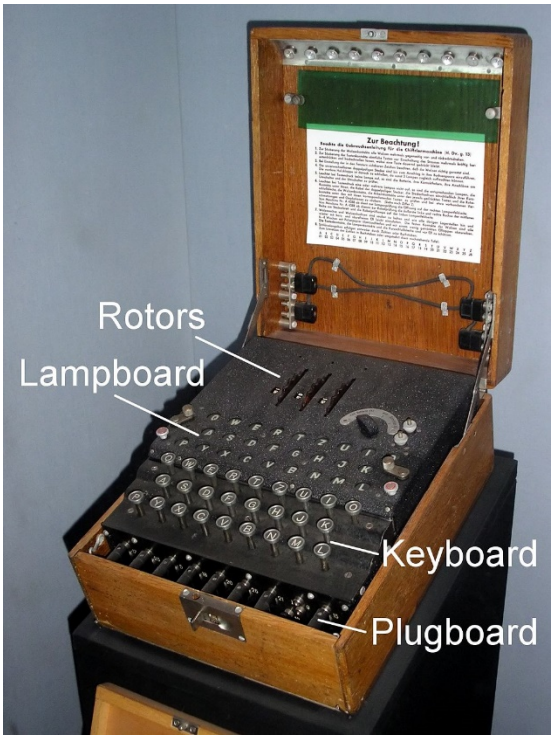
The Enigma Machine

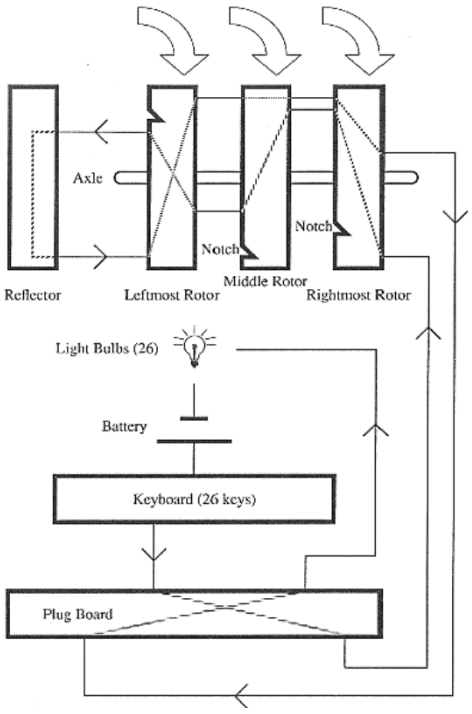
De eerste civiele prototypen van dit elektromechanisch codeersysteem werden reeds begin twintiger jaren gefabriceerd door Chiffriermachinen AG. Latere militaire versies hadden drie rotors met elk daarop 26 elektrische contacten, die de evenzovele hoofdletters van het alfabet door elkaar konden husselen, afhankelijk van de gekozen instelling op elk van de rotors. De afbeelding toont de zogenaamde reflector (met de letter B) aan de linkerzijde, met daarnaast de drie distributierotors (waarbij hier de rotorcode 02-02-16 is gekozen). De 26 contactpunten, voor elke alfabetletter, zijn op de rechter rotor duidelijk zichtbaar.



Hoe werkt dat nou? Mathematisch gesproken wil je een "permutatie" p toepassen op de verzameling {A, B, C,…., Z}. Hierbij is p een afbeelding, die elke letter precies aan één unieke andere letter koppelt, bijvoorbeeld p(D)=G, p(I)=O, p(R)=E, p(K)=S enzovoorts, zodat in dit geval DIRK codeert als GOES. De decodering is eenvoudig, mits men dezelfde sleutel p gebruikt bij een identieke machine (reciprociteit, ofwel $p^{-1}=p$, dus als bijvoorbeeld p(D)=G dan ook p(G)=D). Bij een afwijkende sleutel verschijnt een fout terugvertaald bericht.

Maar hoe was dit permutatieproces in de Enigma geïmplementeerd? In de linker afbeelding zien we een 3-rotor Wehrmacht Enigma [uit het Imperial War Museum, Londen] en rechts het logische elektrische schema [McCurdy School, 2007].





Het belangrijkste deel van het toestel bestaat uit een toetsenbord, een stekkerbord, de eerdergenoemde reflector plus de drie draaiende rotors (voorzien van een stappenmechanisme) en een paneel met lampjes. Het indrukken van een toets wordt via het elektromechanisch circuit vertaald in een oplichtend lampje, dat de vercijferde letter toont. Maar hoe gaat dat dan precies?

Als bijvoorbeeld de D-toets en daarmee een wisselschakelaar wordt ingedrukt, vloeit de stroom vanaf de batterij naar het stekkerbord. Dit stekkerbord maakt het mogelijk (desgewenst) de bedrading tussen toetsenbord en de doorkoppeling naar de rechter rotor te wijzigen, met maximaal 13 kabeltjes. Zou bijvoorbeeld dat de letter D met een kabel zijn doorgelust naar de letter X op het stekkerbord, dan gaat de stroom vervolgens van de uitgaande contactbus op het stekkerbord naar de drie stapsgewijs ingestelde rotors. Die laatsten vercijferen de letter X via hun interne bedrading, om vervolgens weer door de reflector via een andere weg geretourneerd te worden door dezelfde drie rotors, en dan terug naar het stekkerbord. Stel nu dat de stroom op het stekkerbord arriveert bij letter S, maar dat een kabeltje is gelegd tussen S en G, dan gaat de stroom verder naar de *niet ingedrukte* toets G waardoor de gloeilamp G gaat branden. De D wordt dus vercijferd als G.

Overigens, bij het indrukken van de G-toets legt de stroom dezelfde weg af in omgekeerde richting en laat dus het D-lampje branden. Dit is de voor Enigma typerende omkeerbare versleuteling. Merk op dat een letter daarom altijd met een andere letter moest worden gekoppeld, dit om kortsluiting te voorkomen.

Het aantal mogelijkheden dat de Enigma toeliet is verbijsterend: $3x10^{114}$. Dit werd mede mogelijk doordat de rechter rotor één positie doorklikte na iedere toetsaanslag, de middelste rotor één positie doorklikte na 26 toetsaanslagen en de linker rotor tenslotte één positie doorklikte na 26^2 toetsaanslagen. Merk op dat dus twee opeenvolgende aanslagen van dezelfde letter een ander resultaat opleverde. Om dit ongelooflijke aantal permutaties in perspectief te plaatsen: er zijn ongeveer 10^{43} legale schaakposities en ongeveer 10^{80} atomen in het waarneembare universum. Zonder overlopers en spionnen, die de geallieerden belangrijke informatie over de werking van de Enigma bezorgden, was het vermoedelijk onmogelijk geweest om de code te kraken. Maar omdat ze die geheime informatie slim toepasten, hoefden de cryptoanalytici tegen het einde van de oorlog dagelijks 'slechts' ongeveer 10^{23} (preciezer: 107.458.687.327.250.619.360.000) mogelijke combinaties te beschouwen. Dat deden ze met een elektromechanische 'computer', die systematisch alle standen van de drie rotors doorliep, aangedreven door stappenmotors. Die computer was ontworpen door Turing zelf, die hij liefkozend 'Christopher' noemde, naar de jongen waarmee hij op kostschool (meer dan) bevriend was geraakt.

Aanvankelijk bleek de machine niet in staat om ook maar één code te breken. Het doel was echter dit steeds binnen het bestek van een dag te realiseren, omdat de Duitsers voor iedere dag een andere sleutel voorschreven, aan de hand van een vooraf gedistribueerde lijst (sommigen onder ons kunnen zich nog de TAN-codes van de Postbank herinneren, zo ongeveer). Het leek dus onbegonnen werk. Doch essentieel bleek de toevallige *Aha Erlebnis* van Turing dat alle berichten eindigden met de tekst 'HEILHITLER', hetgeen een kapitale blunder van de gezagsgetrouwe Duitsers was. Deze sequentie met zes unieke letters maakte het namelijk mogelijk om de vertaalsleutel van de dag relatief snel te vinden met *smart brute force*. Na *plug board linkage* en vertaling van een bevel of ander bericht konden onder meer de doelen en coördinaten van Duitse duikboten worden afgeleid, waarmee deze vernietigd konden worden voordat ze koopvaardij- en militaire vloten konden doen zinken. Natuurlijk moest dat bijzonder behoedzaam en selectief gebeuren, de vijand moest niet in de gaten krijgen dat de Enigma-codes waren gekraakt! Opnieuw speelde Turing hierin een cruciale rol door de geheime dienst MI6 c.q. defensie ervan te overtuigen dat iedere keer middels statistische analyse een afweging moest worden gemaakt of de winst van een interventie opwoog tegen het risico van ontdekking. Zo kon ook de beste keuze voor de invasie (Normandië) worden bepaald, omdat er inzicht was in de Duitse troepenbewegingen aan de diverse fronten. Al met al hebben de activiteiten van Bletchley Park naar schatting een miljoen slachtoffers bespaard, omdat het beloop van de oorlog met minstens twee jaren werd bekort. Sommige historici gaan zelfs zover dat het kraken van de Enigma-code de Duitsers de overwinning heeft gekost.

The Turing Machine and the Imitation Game

Turing en zijn medewerkers kregen na de oorlog echter geen heldenstatus, want alles moest vernietigd worden of strikt geheim blijven. Hij werd benoemd tot hoogleraar wiskunde in Cambridge en richtte zich op verschillende onderwerpen uit de zuivere wiskunde, maar vooral op wat we nu informatica noemen. De *Turing Machine* was de opmaat voor een strikt formele beschrijving van machine-instructies, wat we nu als besturingssystemen en computertalen kennen. Hij had in het begin van de jaren vijftig zelfs verschillende 'computers' in zijn eigen huis opgesteld, waarmee hij experimenteerde.

In de 2014 *Weinstein Studios* film *The Imitation Game* zien we Alan Turing, (gespeeld door Benedict Cumberbatch) met zijn collega's, waaronder *fiancé-for-a-day* Joan Clarke (Keira Knightley), aan het werk in Bletchley Park. De filmnaam verwijst naar de cynische vraag die Turing opwerpt tijdens latere politieverhoren over zijn homoseksualiteit, of een machine een mens kan imiteren en of je daar spelenderwijs achter kunt komen door de juiste vragen te stellen.

Drie van Turing's collega's in Bletchley Park waren sterke schakers. Hugh O'Donel Alexander, Stuart Milner-Barry en Harry Golombek maakten deel uit van het team dat uiteindelijk de Enigma Code zou kraken. Zij waren in de zomer van 1939 in Buenos Aires gearriveerd om de Schaakolympiade te spelen, maar op 1 september 1939 kwam het bericht van de Duitse inval in Polen, waarna het Verenigd Koninkrijk Hitler de oorlog verklaarde. De Engelsen gingen stante pede met de boot naar huis.

Turing en zijn team (still uit de film 'The Imitation Game')

Onderstaande foto toont een beeld van Alan Turing met een Enigma, gemaakt van een half miljoen stukjes leisteen uit Wales met een totaalgewicht van 1.500 kilogram. Het werd vervaardigd door Stephen Kettle en in 2007 geplaatst in het Bletchley Park Museum, in opdracht van de Amerikaanse filantroop Sidney Frank.

The Paper Machine

In deze sectie, over de schaakcomputeractiviteiten van Turing, citeer ik (onvertaald) de meest relevante delen uit een publicatie van Frederic Friedel and Garry Kasparov, die in 2017 is verschenen onder de titel *Reconstructing Turing's "paper machine",* zie voor het volledige artikel https://easychair.org/publications/preprint/WjKW.

It is an amazing fact that the very first chess program in history was written a few years *before* computers had been invented. It was designed by a visionary man who knew that programmable computers were coming and that, once they were built, they would be able to play chess.

The man, of course, was Alan Turing (sculpture above in Bletchley Park), one of the greatest mathematicians who ever lived. He was very interested in chess, but in spite of having a brilliant intellect and putting a lot of effort into learning the game he remained a fairly weak player. While working at Bletchley Park on breaking the German Enigma code (and probably decisively influencing the outcome of the War) he took chess lessons from his GM-strength colleagues in the decryption team, and even invented "Round-the-house chess" (after each move you had to run around the house, and could play two moves in a row if you overtook your opponent). Turing was an Olympic-class runner, and this was his way of balancing out his lack of chess talent.

In 1948 Turing, assisted by his friend David Champernowne, wrote the instructions that would enable a future machine to play chess. They called the program Turochamp, but it popularly became known as "Turing's paper machine." Turing's goal was to "make a machine which would play a reasonable good game of chess, i.e. which, confronted with an ordinary chess position, would after two or three minutes of calculation, indicate a passably good legal move." The program introduced the concept of quiescence. Champernowne wrote: "Most of our attention went to deciding which moves were to be followed up. Captures had to be followed up at least to the point where no further capture was immediately possible. Check and forcing moves had to be followed further. We were particularly keen on the idea that whereas certain moves would be scorned as pointless and pursued no further, others would be followed quite a long way down certain paths."

Turing (and Champernowne) used the following piece values: p=1, N=3, B=3 ½ , R=5 and Q=10. In addition they had the following positional evaluation functions:

(1) Mobility. For the Q,R,B,N: add the square root of the number of moves the piece can make; count each capture as two moves.
(2) Piece safety. For the R,B,N: add 1.0 point if it is defended, and 1.5 points if it is defended at least twice.
(3) King mobility. For the K, the same as (1) except for castling moves.
(4) King safety. For the K, deduct points for its vulnerability as follows: assume that a Queen of the same colour is on the King's square; calculate its mobility, and then subtract this value from the score.
(5) Castling. Add 1.0 point for the possibility of still being able to castle on a later move if a King or Rook move is being considered; add another point if castling can take place on the next move; finally add one more point for actually castling.
(6) Pawn credit. Add 0.2 point for each rank advanced, and 0.3 point for being defended by a non-Pawn.
(7) Mates and checks. Add 1.0 point for the threat of mate and 0.5 point for a check.

At the time there was no machine, of course, that could execute this set of instructions, and so Turing acted as a human CPU, using paper and pencil, requiring more than half an hour to calculate each move. He played a game against Champernowne's wife, who was a beginner at chess, and won. Then he played one against a colleague, Alick Glennie, which he lost. This second game was recorded:

Turing's paper machine – Alick Glennie, Manchester 1952:

1. e4 e5; 2. Nc3 Nf6; 3. d4 Bb4; 4. Nf3 d6; 5. Bd2 Nc6; 6. d5 Nd4; 7. h4 Bg4; 8. a4 Nxf3+; 9. gxf3 Bh5; 10. Bb5+ c6; 11. dxc6 0-0; 12. cxb7 Rb8; 13. Ba6 Qa5; 14. Qe2 Nd7; 15. Rg1 Nc5; 16. Rg5 Bg6; 17. Bb5 Nxb7; 18. 0-0-0 Nc5; 19. Bc6 Rfc8; 20. Bd5 Bxc3; 21. Bxc3 Qxa4; 22. Kd2 Ne6; 23. Rg4 Nd4; 24. Qd3 Nb5; 25. Bb3 Qa6; 26. Bc4 Bh5; 27. Rg3 Qa4; 28. Bxb5 Qxb5; 29. Qxd6 Rd8; 0-1.

In the early 1950s Turing began programming Turochamp on a Ferranti Mark 1 computer (above, image courtesy of Chess Programming Wiki) at the University of Manchester. At the same time he tried implementing a rival program, Machiavelli, by Donald Michie and Shaun Wylie on the same machine. Unfortunately he had not completed the task before his tragic death in 1954.

Interestingly nobody seems to have written an actual computer program using Turing's algorithm for more than fifty years after Turing devised it. In 2004 the programming team at the ChessBase software company decided to create a chess engine based on Turing's algorithms. The basis was Turing's description and the sample game against Glennie, and the programmer who undertook the task was Mathias Feist. He built a standard chess program interface for the ChessBase Fritz program, so that the Turing engine could be tested.

In the process, however, the ChessBase team encountered a problem: the engine refused to duplicate all of Turing's moves as recorded in the Glennie game. It deviated in ten places – significantly already on the first move: the Turing engine wanted to play 1. e3, whereas Turing had executed 1.e4 in his 1952 game.

The ChessBase team spent some weeks re-examining the Turing papers and discussing the implementation in the Turing engine. Since they could not locate the error they consulted Ken Thompson, one of the pioneers of computer chess (and author of Unix and C). He spent some time wrestling with the problem, and in fact wrote his own code on the basis of his reading of the Turing instructions. His program produced most of the ChessBase deviations from the Glennie game. It was quite baffling.

So Frederic Friedel contacted Donald Michie, who had been with Turing at Bletchley, describing the problem in general and the most significant deviations the ChessBase Turing program produced after the prescribed three-ply search (which Turing used in his paper-and-pencil game). "It is possible that we have got something wrong" he wrote, "but I doubt it, since very often, especially in the beginning, we get the same moves with exactly the same values. I think it is possible that Turing tired after fifteen moves, when in addition the position became quite complex?!"

Michie's reaction (in a phone conversation) was: "You are looking for a bug in the program, Frederic? No, no, you should look for it in Alan Turing! Alan did not care about details; he was interested in the general principle." He also pointed us to a quote by Champernowne, who had assisted Turing during the 1952 game: "In the actual experiment I suspect we were a bit slapdash about all this and must have made a number of slips since the arithmetic was extremely tedious with pencil and paper."

Bletchley Park

In June 2012 Friedel travelled to Manchester to attend the Alan Turing Centenary Conference. On the way he spent June 23rd, Turing's 100th birthday, at Bletchley Park, a visit he will never forget. In Manchester he assisted Garry Kasparov in a commemorative lecture on Turing's chess involvement. The two spoke about the reconstruction of the paper machine, and Kasparov actually played a game against it. In the lecture Kasparov pointed to a key moment in the game: "It was the first time that a machine played a game of chess... After a pretty average game they reached an equal position and the machine blundered a queen – taking the pawn (with 29. Qxd6), after which the queen is pinned (Rd8), and White resigned."

Kasparov: "Blundering a queen in one move, that is not a great accomplishment. But still the game was played and there were 29 moves, not just legitimate moves but you may call them decent moves." He confirms that he will play a game against the reconstructed engine at the end of the lecture, "the first official display of the Turing machine playing against a professional player. ... I think that at five seconds a move, it could beat most amateurs..."

In the lecture Friedel discussed the deviations we were confronted with and how they probably came about. An important point is that Turing, working with paper and pencil, clearly used his intuition to come up with the moves. And without knowing it he used Alpha-Beta pruning, a technique that was invented by Allen Newell and Herbert A. Simon half a decade later, when machines could actually execute code. Essentially Turing did not bother to calculate every single move in a position. Some he thought were clearly inferior, and it seemed pointless to spend a lot of time working out exactly how bad they were (which is the point of Alpha-Beta).

In the lecture we discussed the problem of the very first move: Turing played 1.e4, the ChessBase engine (and Thomson's reconstruction) come up with 1. e3.

In the initial position material is equal, no pieces are taken, nothing has happened yet. There are twenty legal moves for White – each of the eight pawns can move up one or two squares, and the two knights have two moves each. One wonders if Turing made 20 calculations, each taking a number of minutes, before he executed the first move, 1. e2-e4. It is one of the two most common moves that start a chess game (the other is 1. d2-d4).

Let us apply Turing's rules to the initial position. The e-pawn gets 0.4 points for moving two squares forward, but just 0.2 points for moving one step (rule 6: pawn credit). So 1. e4 is better. But then we apply rule 4 (king safety) by placing a queen on the square e1, counting the number of squares the queen can move to – they are the squares from which the king could be attacked by a rook or a bishop – and taking the square root to define its mobility. 1. e2-e3 gets a malus of 1 (square root of one square, e2, to which the queen can move), whereas 1. e2-e4 reduces the positional advantage by about 1.4 points (square root of 2, for e2 and e3).

In his game score Turing gives 4.2 points for the move he played, 1. e2-e4, and adds an asterisk to the value, which indicates that "every other move has a lower position-play value." The ChessBase engine, and that of Ken Thompson, give 1. e2-e3 a positional value of 4.40, compared to Turing's move: 0.2 less for advancing just one pawn but 0.4 more for king safety. Mathias Feist wrote "Turing's calculation for e2-e4 is correct, which means he did not really calculate e2-e3. As a chess player he knew that this move is not logical, because in the initial position king safety doesn't play a role. He probably looked at it and thought: everything is the same for both moves, except e2-e3 is 0.2 points worse because the pawn moves just one square. So he discarded it without further calculation."

Incidentally the move 1. d2-d4, one of the most popular in tournament chess, was certainly calculated by Turing, who must have seen that it is clearly worse. Putting a queen on the king square shows four squares to which it can move, i.e. the malus for king safety is 2.0.

Let us take a look at move four: Turing played 4. Ng1-f3 and gave a value of 2.0. The ChessBase Turing engine plays 4. d4xe5 and give it a value of 1.1. Why?

After 4. Ng1-f3 the mobility of the queen on d1 sinks from six to three squares, and consequently the positional value from 2.4 to 1.7 (= -0.7). The mobility of the knight g1 increases from three to five squares, and so the score increases from 1.7 to 2.2 (= +0.5). The mobility of the rook on h1 increases from 0 to 1 square, improving the score from 0.0 to 1.0 (= +1.0). So 4. Ng1-f3 improves the evaluation of the position by 0.8 points.

After 4.d4xe5 the d4 pawn has advanced by one square (+0.2) but is no longer defended (-0.3). The mobility of the queen increases from 6 to 9 squares (+0.6). The black pawn on e5 has disappeared, and with it the bonus for the two rows it had advanced (+0.4 for White). The black king safety increases from 4 to 5 (+0.2 for White). So in total 4. d4xe5 increases the score by 1.1 points.

There were also some problems with the quiescence search ("captures had to be followed up at least to the point where no further capture was immediately possible"), as applied by Turing in the Glennie game.

On move five Turing played Bc1-d2 and gave a value of 3.5. But after 5. Bd2 exd4; 6. Nxd4 Bxc3; 7. bxc3 (or Bxc3) Nxe4 no more captures are possible and White has lost a pawn, while after 5. Bg5 exd4; 6. Nxd4 Bxc3; 7. bxc3 (or Bxc3) Nxe4 White plays 8. Bxd8 Kxd8 and in the resulting quiescent position White has a queen (10 points) for a knight and pawn (3+1).

Similarly in the above position Turing played 15. Rh1-g1 and gave it a 1.2 score. But the quiescent search reveals that 15. Rg1 Bxf3; 16. Qxf3 Qxa6 loses a pawn for White, which 15. Rh1-h3 does not.

There are a number of other deviations in the game, more towards the end, when Turing was apparently tiring. The game clearly demonstrates how tedious and error-prone human calculation in such a situation can be. On a modern computer the ChessBase Turing engine needs less than five milliseconds to calculate the values for all legal moves in a given position, and to choose the best one to play. Turing took fifteen to twenty minutes.

The Killing Machine

"L'Etat c'est moi", zei Lodewijk XIV, of zeiden althans zijn tegenstanders spottend. En met mijn onderdanen doe ik zoals het mij goeddunkt, ik kan ze verheffen of vernietigen. Nou kun je zeggen, *sois*, de donkere Middeleeuwen zijn maar nauwelijks voorbij in het Frankrijk van rond 1700. Maar wat te denken van Groot-Brittannië in het begin van de jaren vijftig van de verleden eeuw, 70 jaar geleden? Een gidsland met een verlichte parlementaire monarchie, een leider van de vrije wereld die de dictator versloeg, toch? *Britain leads! Indeed, but not quite.*

Er was ingebroken in januari 1952 bij Turing thuis, met veel schade. Hij deed aangifte van die inbraak, maar de politie vond het vreemd dat er niets was gestolen en Turing zich excentriek en wat arrogant gedroeg. Na uitgebreid politie-onderzoek, waarin Turing werd geconfronteerd met zijn tegenstrijdige verklaringen, liet hij zich weinig diplomatiek ontvallen dat het zijn 19-jarige vriend Arnold Murray was die de ravage had aangericht. Dat was de opmaat tot een ongekend harde vervolging en een schokkend proces wegens *"Gross indecency contrary to Section 11 of the Criminal Law Amendment Act 1885".*

Hij mocht van de rechter uiteindelijk kiezen tussen twee jaar gevangenisstraf óf een experimentele chemische castratie, teneinde van zijn indecent gedrag voorgoed te worden genezen. Hij koos voor het laatste omdat hij anders zijn 'Christophers' voorgoed zou verliezen. Maar toen zijn voormalige verloofde maar nog steeds innig bevriende Joan Clarke hem na de veroordeling weer kwam opzoeken, trof zij tot haar ontzetting een fysiek verzwakte man en een gedesoriënteerde schim van het genie aan. Haar smeekbeden om met de advocaten in overleg te gaan, wees hij af.

Een jaar later, op 7 juli 1954, pleegde Alan Turing (zeer vermoedelijk) zelfmoord door een hap uit een met cyanide vergiftigde appel te nemen. Een hardnekkig gerucht wil dat dat (de Beatles-fans) Jobs en Wozniak heeft geïnspireerd tot het eigen Apple-logo.

In december 2013, op voordracht van premier Gordon Brown, verleende koningin Elizabeth Alan Turing (vermoedelijk op basis van zijn verdiensten voor het vaderland die vijftig jaar na dato openbaar waren gemaakt) postuum gratie en werd zijn veroordeling uit de boeken geschrapt. Maar tussen 1885 en 1967 waren maar liefst 45.000 mannen veroordeeld voor '*sodomy*' en '*indecency*', onder wie Oscar Wilde. Wat is er sodeju met díe vonnissen gedaan?