



## The Maths of Board Games

Professor Sarah Hart

10 October 2023

Why are there chess Grandmasters, but not Grandmasters of noughts and crosses (otherwise known as tic-tac-toe)? It is because chess is “harder” – but what do we really mean by that? Answering that question leads us to develop the idea of mathematical complexity, which is a measure of how ‘big’ a game is. We’ll look at the complexity of popular board games, and ask: What is the hardest game of all time?

### A brief history of board games

Play is part of being human, and archaeological evidence of games dates back many millennia. There are ancient games that endured for hundreds, if not thousands, of years; a handful are still played today. I’ll make the case that there are good mathematical reasons why. Some of the oldest gaming pieces are dice, but people also used stones, sticks, and bones. There are many types of games – card games, dice games, and even balancing games like Jenga and others. We’ve already talked a bit about cards and dice in the context of gambling and probability in a previous lecture of mine (link at the end of the transcript). So, we are going to focus exclusively today on board games. Historically, specially constructed gaming boards were probably the preserve of the wealthy, and we have few extant examples, but that doesn’t mean the rich were the only ones to play – it’s easy to mark out a “board” in the sand or on the ground or scratch one on a nearby surface like a stone wall or a wooden table. Games were played for many reasons, too. Entertainment is one, of course, but it could also have religious, moral, or spiritual significance, or be played as a way to practice strategic thinking – a mini war waged between opposing forces. I’ll show you a few of my favourite examples as we go.

### Characteristics of a game

In exploring the maths of board games, it makes sense first to think about how a game is put together, and what goes into the mix to give each game its particular playing experience. I want to draw your attention to three measures, or characteristics, of a game that will feed into the playing experience, each of which we can imagine as a continuum.

1. Number of rules. The rules of a game specify the board, pieces, initial set-up, possible moves, and of course the definition of a win. These can range from very simple to very complex. At the simple end, we have noughts and crosses, whose rules can be encapsulated very briefly. Two players take turns placing counters (noughts and crosses) in a  $3 \times 3$  grid. The first player to make a line of three wins (if neither player does, it’s a draw). Another simple game is *domineering*. It’s played usually on a chessboard, but any size grid can be used. Player 1 places dominoes horizontally, Player 2 places them vertically. The winner is the last person able to place a domino. In the middle of the scale are games like backgammon, and chess. There are some incredibly complex games out there. The most extreme I’ve been able to find is called *The Campaign for North Africa*, which simulates part of the action in the Second World War. Each turn, you have to do a huge number of things, from feeding your troops to raiding Malta. There are rules covering every tiny aspect. For instance, you have to ensure that your troops have fuel. You lose a bit due to evaporation every turn. For every nationality’s army, it’s the same evaporation rate, except for the British. Their evaporation rate is higher because, in the actual war, they used large fuel drums instead of jerry cans! The board is nearly a metre wide and over three metres long (34 by 115 inches). The estimated time for one game is 1500 hours, which, if you give up your job and play for 35 hours a week, will take you 43 weeks. I work hard to prepare for these lectures. But I have not played this game.

2. Number of players. There's no upper limit here, although in practice it's hard to get more than a few players round a board. Monopoly has pieces for up to six players, and there's an ancient game called "astronomical chess" (though it has nothing to do with chess really) that has room for up to seven on its heptagonal board. But most boards are square, and therefore many games are designed for up to four players. Ludo, or, equally, its traditional Indian antecedent Pachisi, is just one of many examples of this. Chinese checkers (which are neither Chinese nor have anything to do with chequers/draughts, having been invented in Germany in 1892) have a board arranged in a six-pointed star and work quite well for three players because of this. The majority of traditional board games are for two players, including chess, backgammon, draughts, and ancient games like Nine Men's Morris, Ur, Mancala, and so on. There are even 1-player games – solitaire, for example, where the objective is to remove all but one counter from the board. You might think a game can't have zero players, but the *Game of Life*, invented by mathematician John Conway, can be considered in this category. It's played on a grid which can extend as far as needed in all directions. To set it up, you fill some of the squares/cells (either at random or however you wish). These are "live". The remaining cells are "dead". Then the game plays itself according to very simple rules. Any cell has exactly eight neighbouring cells. At each "turn", a live cell with zero or one live neighbour dies as it's too isolated. A live cell with more than three neighbours dies due to overpopulation (not enough resources). Otherwise, it survives. Meanwhile, any dead cell with three live neighbours becomes a live cell, thus spreading the population. Otherwise, it stays dead. The possible outcomes of this game are extremely interesting! You can make "spaceships", logic gates, computers, and many other fascinating objects.
3. Chance versus skill. Some games involve absolutely no decision-making at all, and thus no skill. They are purely based on chance. One example is *Snakes and Ladders*. At the other extreme we have games like chess, or (if you allow a vertical board) Connect Four, where no luck is involved – it's 100% skill. In between these extremes are games like *Backgammon* where there are dice rolls and therefore a component of chance.

For the rest of today's talk, we are going to stick to two-player games, and mainly focus on games of pure skill (and in addition, we will make a technical assumption that both players have full knowledge of the state of the game, unlike games such as *Cluedo* (*Clue* in the U.S.) where different players have different information). We'll also stick closer in the difficulty scale to the Noughts and Crosses end than the *Campaign for North Africa* end!

## Measures of Complexity

Games of "pure skill" are not necessarily difficult. *Noughts and crosses* is a game of pure skill but it's not very difficult to master. This is where we need another measure, some way to understand the complexity of the game. One approach is that we can give an estimate of the "size" of the game. One measure of size is to ask how many possible configurations we can see during gameplay. The idea here is that with a "small" game with not many possible configurations, or "states", over time one could learn them and get familiar with what works in each configuration. This measure is called the *state-space complexity*.

I'll give a couple of examples. For *noughts and crosses*, we can estimate the state-space complexity quite quickly. Each square can either be empty or contain a nought or a cross. Three possibilities for each square, nine squares, so  $3^9 = 19,683$  possible states. But, of course, this is an overestimate because you can't have (for instance) nine crosses. Conventionally crosses start, so at any point you either have the same number of noughts as crosses, or you have one extra cross. You can work out the number of possibilities after, say, two moves: there are 9 positions for the cross and 8 for the nought, so  $9 \times 8 = 72$  game states after two moves. If you do this for every possible number of moves the total comes to 6,045 (or 6,046 if you include the initial empty board configuration). Even this is an overcount because it still includes some configurations that cannot arise in the game, such as states where there is both a line of crosses and a line of noughts. The true number of so-called "legal" positions can be calculated exactly: It is 5,478. And in fact, if you consider rotations and reflections of the board as things that don't affect the essential situation, this reduces even more to 765. Another fairly simple case is *Connect Four*. Here you play on a vertical board, taking turns to drop counters (traditionally red versus yellow) into the board; the first to make a line of four counters wins. The board has seven columns each holding up to six counters. If you have, say, three counters in a column, then each can be red or yellow, so there are  $2 \times 2 \times 2 = 8$  possibilities. This means the total number of possibilities (ignoring the fact that some, like having five consecutive ones the same colour, are impossible to occur in any game because we stop as soon as one player has a line of four), for a given column, is  $1 + 2 + 4 + 8 + 16 + 32 + 64 = 127$ . By my calculation, there are 16 illegal columns – for example, YRRRRY

(where Y is yellow and R is red), or YYYYYY. This brings it down to 111 legal columns, and there are seven columns, so there's an upper bound of  $111^7 = 207,616,015,289,871$  for the equation the size. This is still a considerable overestimate because, of course, it doesn't take into account horizontal or diagonal lines of counters that might result in illegal configurations.

The actual number of legal positions has been calculated precisely, with machine assistance. It's 4,531,985,219,092 (a factor of about 45 smaller). And, as we'll see, with any game even slightly more complex than these simple examples, not only is the exact number pretty soon impossible to calculate, but it's enough to know a rough bound – we usually write the state-space complexity to the nearest power of ten. So, for *noughts and crosses*, that's  $10^3$ . For *Connect Four*, it's  $10^{13}$ . We'll worry about chess later!

The second measure of size is obtained by trying to count all the possible games. Here, we imagine constructing a “tree” where from each node there are branches representing every possible move. The complete game is then mapped out by this tree, which contains within it all possible games. In *noughts and crosses*, the first player (crosses) can do essentially three things: centre, corner, or edge. Then *noughts* can respond in a number of ways to each of those, and we draw in all of these nodes to construct the full tree. Even for *noughts and crosses*, this gets quite big, and we certainly wouldn't want to draw out the whole tree. One example where we sometimes can is domineering, as long as we choose a sufficiently small board. If we play on a  $3 \times 3$  board, then Player 1 has essentially two first moves: cover a corner or cover the centre (plus an adjacent square). Anything else is a mirror image or rotation of one of these. The rest of the game tree can then be completed, again treating positions that are reflections or rotations of each other, as long as they preserve horizontal and vertical, as equivalent. Every path through the tree to an endpoint (endpoints are called leaves, to continue the tree analogy) represents a different possible game. The game tree size is the number of leaves in the game tree; that is, the number of possible games that can be played. We can only draw the game tree for very small games. For  $3 \times 3$  domineering the game tree size is 11. But this grows rapidly for larger boards!

There's one final measure called the game tree complexity, which is an attempt to measure how much information is needed in order to determine the outcome of the entire game from the initial position – in other words, assuming both players are playing optimally, who will win (or will it be a draw)? If we have the whole game tree, we can label each leaf/final node according to whether it's a win for Player 1, a win for Player 2, or a draw. We can then work backwards labelling each node (assuming each player plays optimally) so that ultimately, we can get back to the very start and “solve” the game, in other words, determine the outcome (win, lose, draw) for each player. But to do this we don't always need the whole game tree. For example, in *noughts and crosses*, if at any point you can get three in a row, of course, you will, and so the game will end at the next step. So, strictly speaking, we don't need the whole theoretical game tree, we just need enough of it to get us to a point where, assuming both players play optimally, the game will finish, and we'll know the outcome. Another step back might be if we are in a position where our opponent has two lines that can be completed on their next turn. Of course, we can only prevent one of them. So, we know that we will lose – at least if our opponent is concentrating! The game tree complexity is the number of leaves in the smallest possible part of the full game tree that allows us to fully determine the outcome of the game. Usually, both game tree size and game tree complexity are so big that we have to estimate them both and we tend to use the same estimates, so we will mostly conflate these terms and just talk about game tree complexity. Of the games we've discussed, *noughts and crosses* have game tree complexity of about  $10^5$ , for *Connect Four* it's  $10^{21}$ , and for  $8 \times 8$  domineering it's  $10^{27}$  – all these are approximations.

## Strategies and combinatorial game theory

All of these complexity estimates are interesting, but what can we learn from them? One thing we tend to want to do when playing a game is win. One might imagine that the game tree complexity will give us a clue as to how hard it might be to find a strategy (and potentially program a computer to play the game as well). If we know the full game tree, we can map out a strategy for the game by picking the best moves at each stage. For example, in  $3 \times 3$  domineering, if Player 1 has put a horizontal domino along one edge, then Player 2 should put a vertical domino through the centre as this will win the game. On the other hand, if Player 1 places their domino through the centre, then whatever Player 2 does we can see from the tree that Player 1 will win on their second move. Thus, we have a strategy: If you can, place your domino so that it covers the centre square; otherwise move randomly. This means that  $3 \times 3$  domineering is a “solved” game, in that we know the outcome (assuming both players play optimally). These ideas (and many others that we don't have time to discuss) come into the area of mathematics called “combinatorial game theory”, to distinguish it from the kind of game theory used in economics, that covers things like how to bid at auctions

and so on. There are actually several levels of “solved”. A game is strongly solved if there’s an algorithm that gives an optimal strategy from any position, even if one or more turns have been taken. *Noughts and crosses*, and several small grid versions of *domineering*, are strongly solved. If a game is weakly solved, it means you have a strategy that works from the beginning of the game (in other words, if Player 1 can guarantee a win, or failing that a draw, the strategy is known, if Player 2 can guarantee a win whatever Player 1’s first move is, then that strategy is also known). *Connect Four* was solved in 1988 by James Dow Allen and Victor Allis. Initially, it was weakly solved, but since then computing power has increased to the extent that it’s now strongly solved. If both players play perfectly, Player 1 can win in at most 41 moves – the first move is to place a counter into the centre column. If Player 1 instead puts their counter in a column adjacent to the centre, then Player 2 can force a draw, but can’t guarantee a win. If Player 1 does anything else, Player 2 can win. Finally, if a game is ultra-weakly solved, it means you know theoretically whether the outcome will be win, lose, or draw for each player (assuming optimal play from both players) but don’t necessarily have a strategy. One example is “*Chomp*”, which is played on a rectangular grid (or chocolate bar!), with players taking turns to “eat” a square, at which point they must eat all remaining squares below and to the right of the chosen square. As we are all so polite, the person who is forced to eat the last square loses – it’s not possible to draw. Were there to be a winning strategy for Player 2, it would include a winning response to Player 1 eating just the bottom right square. But if you are Player 1, you can just play that as your first move, and so guarantee a win. This is known as “strategy-stealing”. We have no idea what the strategy is, except that there must be one. So, *chomp* is ultra-weakly solved. One game we can’t even say that about is the one we will discuss next: chess.

## Chess and Computers

Chess has a very long history with roots in India dating back at least 1500 years. There are many variants and related games, two important examples being versions popular in China (*xiangqi*) and Japan (*shogi*). I’m going to focus exclusively on the form of chess played in Europe from the 15<sup>th</sup> century, whose rules were standardized about 150 years ago – it’s sometimes called “International chess” or “Western chess” to distinguish it from other games. I’ll just say chess. We don’t have grandmasters of *noughts and crosses*, but we do have grandmasters of chess, and the reason is that it’s a lot harder to be good at chess! How complex, then, is chess?

An early mathematical paper on board games was a 1950 article by the American mathematician Claude Shannon, called “Programming a Computer for Playing Chess”. It was theoretical in that it didn’t actually suggest a specific computer program – instead, Shannon estimated what we now call the game tree complexity of chess to show that it’s completely infeasible to solve chess by brute force. It’s a very nicely written paper which is available online – I’ve put a link in the transcript, or you can easily find it by searching. With the vast computer power we nowadays have, we can do a lot more by brute force than was possible then, so tighter bounds, or even exact numbers in some cases, are now known, but they do not affect the essential thesis, which remains true now – a complete game tree for chess is beyond the capacity of any computer we know how to build. However, computers have become very good at chess, and interest in chess-playing machines has been high for a long time. Longer, perhaps, than we might think.

Between 1770 and 1854, a chess-playing automaton known as The Mechanical Turk wowed audiences across Europe and America, beating many illustrious opponents, including Benjamin Franklin and Napoleon Bonaparte. People were suspicious that a machine could really be capable of such feats. Edgar Allan Poe wrote an essay about it, called “The Maelzel Chess Automaton”. He argued that since the machine sometimes lost, it clearly couldn’t be a machine after all, but must be a trick – it had to be that a human was playing, hidden in the table that contained the mechanism. This logic is of course incorrect, although the conclusion was right – the Turk was indeed a hoax, which involved an expert chess player operating it using levers. However, even as early as 1912 there was a genuine machine invented that could play a particular kind of end game, where one side (the machine) has just a king and a rook, and the opponent has just a king. *El Ajedrecista* was built by Leonardo Torres Quevedo and caused a sensation when it was first exhibited in Paris in 1914. There is a known strategy for winning in this very special case, and the machine could therefore be constructed to implement it.

In his paper, Shannon estimated the game space complexity by saying that on average there are about 30 possible options for a player to choose between (this will decrease right at the end of the game if you have just a few pieces left, or are constrained by being in check or similar, but is a good estimate for the greater part of the game). Records of chess games also indicate that a typical chess game between good players lasts on average 40 moves (bad players tend to fight to the death when they should resign, so their games

last longer – I should know, being a bad player myself). Here, by the way, by “move” it’s usual in chess to mean one turn for white and then one for black, while in other games there may be other conventions. Because of this, in mathematical analyses, we use the word “ply” for a “turn” by one player. So, in chess, a move consists of two plies, one for white and one for black. Shannon used 1000, or  $10^3$  as an approximation for the number of possible “white then black” moves. Forty of these in a typical game means that the game tree complexity is  $10^{3 \times 40} = 10^{120}$  (this is nowadays called the Shannon number). The fastest supercomputer even today can “only” do a quintillion calculations per second – that’s  $10^{18}$ . So, if the computers on Earth, it, and every chessboard in existence, will be engulfed by the red dwarf phase of the Sun far sooner than it can complete the analysis.

Having some kind of database where the best move for each legal position is stored is also infeasible, says Shannon. He estimates the number of legal positions (the state-space complexity) by looking at the number of ways to put all 32 pieces on the board (this ignores that sometimes they aren’t all on the board). So, the number Shannon estimates is  $\frac{64 \times 63 \times \dots \times 33}{(8!)^2 \times 2^6}$ . To see why, there are 64 places for the first piece can be, 63 remaining spots for the next, and so on, down to 33 places for the final piece. But the two white rooks are interchangeable, so we have double counted because of that. Also interchangeable are the two black rooks, and the pairs of knights and bishops on each side (we also ignore the fact that each bishop stays on the same colour square). Each side also has eight interchangeable pawns, and that is what the  $(8!)^2$  is doing in the fraction. The outcome is an estimate of approximately  $43^{or}$  for the state-space complexity. The best recent estimates are within a factor of 100 of this number, so it’s not bad. But even if you could store a position on a single atom of silicon, I’ve calculated that atoms of silicon weigh about 600 trillion tonnes. So, it’s not going to happen.

In the absence of a complete solution, what could a chess computer do? One option is to think a few moves ahead – not forty moves. Calculate all possible sequences of, say 3 moves (6 plies). As an indication of the scale of the problem, from the beginning of a chess game, it’s been calculated that the number of possible games three moves in is 119,060,324 (of which 10,828 have resulted in checkmate, amazingly!). So, the computer calculates these and then ranks them based on some function. For example, this could be based on a value rating of the pieces – pawn is 1, knight and bishop are 3, the rook is 5, the queen is 9, and of course the king is vital, perhaps  $\infty$ . There are also more complicated things to encode, like how exposed the king is. But however it is done, the computer can select a move that is likeliest to result in the strongest position. One could also incorporate databases of chess openings and strategies for endgames.

One issue is the so-called horizon problem, where a brilliant sequence of moves, such as sacrificing your queen to checkmate two moves later, or any other temporary setback on the way to a spectacular reversal of fortune, cannot be detected by whatever value function is being used. A human player does not follow all leads to the same distance, rather they think through some promising moves to a greater number of moves ahead while discarding “obviously” poor options. An algorithm called “quiescence searching” began to be used in the 1980s and 1990s. It assesses whether a given situation is in a state of flux; if so, it analyses more moves. If not, it stops sooner. This means with the same amount of processing power the algorithm can see further ahead when it needs to. It’s not perfect but it does achieve much better results than brute force search of the entire tree a fixed number of moves ahead. Such has been the strides since the 1950s that nowadays even chess apps for your mobile phone can beat all but the best players. In the late 1990s the supercomputer Deep Blue beat reigning chess world champion Garry Kasparov, and nowadays computers can reliably beat players of any standard.

## The hardest game of all time?

Chess, by any measure, is highly complex. There is no immediate prospect of solving it. But is it the hardest game? If we agree that by “hard” we are going to mean “high game tree complexity”, then we can rank a few games. *Draughts* has a game tree complexity of  $10^{40}$ . *Nine Men’s Morris* is  $10^{50}$ . Both these games are weakly solved – there is an optimal strategy and if both players use it, the outcome will be a draw. These are both “easier” than chess. But there are harder games than chess out there. If we restrict ourselves to games that plenty of people play, rather than, for instance, infinite chess (yes, this actually exists), then a strong contender for the hardest game of all time is *Go*.

Much beloved in university maths department common rooms the world over, *Go* is played on a square grid. Usually it’s a  $19 \times 19$  grid (that is, 19 lines by 19 lines, so 18 squares by 18, but you can play on any size you like, and it’s recommended to begin on a  $9 \times 9$  grid (that is, a chessboard-size board, which you likely already have) until you learn the ropes. Each player has an unlimited supply of counters, called stones,

usually black and white, with (unlike chess) black moving first. When it's your turn, you place a stone of your colour at a vacant grid point (not inside a square, but at an intersection of lines). The aim is to capture territory on the board by surrounding it with stones of your colour. You can also take "prisoners". If you completely surround one or more enemy stones, you capture them (you aren't allowed to "capture" your own stones or to place stones inside your opponent's captured territory, so your choice of moves is not absolutely free). There are a couple of other rules, for instance, the *ko* rule (from the Japanese word for *eternity*) prevents an infinite loop by forbidding certain configurations. At any point, you can pass at the cost of giving your opponent a stone. If both players pass, then the game ends. When the game ends, each player counts up the vacant points inside their captured territory, plus the number of prisoners they have. Whoever has the highest total wins?

With the standard board, the state-space complexity is extremely high. A rough estimate, given that there are  $361 = 19^2$  points on the board and each can, in theory, be black, white, or empty, means that there are  $3^{361}$  potential configurations, or about  $1.7 \times 10^{170}$ . This is, both literally and figuratively, many orders of magnitude more complex than chess. (The  $9 \times 9$  version has state-space complexity  $3^{81} \approx 4 \times 10^{38}$ , which is slightly less than chess.) How about the game tree complexity? This has been estimated at a whopping  $10^{360}$ . A game takes on average 150 plies, and the number of possible moves each time (known as the *branching number*) averages at 250. These, again, are much greater than chess. So, how do computers fare in this vast game space? It took machines a lot longer to get good at *Go* than it did for them to get good at chess. But in 2016, an AI called *AlphaGo* crushed the then-world *Go* champion Lee Sedol in a 4-1 rout. It looked like it was all over. However, in February 2023, the humans got their revenge. The Financial Times reported that a *Go* player called Kellin Pelrine managed to beat a top *Go* program, and not just once, but in fourteen of the fifteen games they played. He did it by exploiting exactly the kind of horizon problem situation that has been a challenge for chess programs – he used a very long-term strategy that the machine didn't see coming until too late. So, it's not quite all over in the battle of man versus machine!

## Next time...

I hope you've enjoyed this mathematical exploration of board games. It's been the first lecture in my series "Games, Puzzles, Paradox, and Proof". In our leisure time, we like to use our brains, and board games are just one way we've devised to do that. In my next lecture, I'll explore the mathematics behind a certain kind of puzzle that millions of us tackle every day – the Sudoku. I'll show you its links to other kinds of number grids, like magic squares and so-called Latin squares, which have been studied for centuries, and we'll see some applications in areas as diverse as experiment design, coding theory, and even literature.

© Professor Sarah Hart 2023

## References and Further Reading

Three highly rated books for a deeper dive into combinatorial game theory, including loads I didn't have space to get into here, are:

- *Lessons in Play: An Introduction to Combinatorial Game Theory*, by Michael Albert, Richard Nowakowski, David Wolfe, Second Edition 2023, CRC Press, ISBN 9781032475660.
- *Winning Ways for Your Mathematical Plays*, Second Edition, by Elwyn R. Berlekamp, John H. Conway, Richard K. Guy, Second Edition 2001, CRC Press, ISBN 9781568811307
- *Games of No Chance*, by Richard Nowakowski, Cambridge University Press, 2010, ISBN 9780521646529.

There are many online resources out there. Here are some to get you started.

- There's some discussion of dice games and card games as part of the discussion on probability in my Gresham Lecture *Lottery-Winning Maths*: <https://www.gresham.ac.uk/watch-now/lottery-maths>. You might also like *The Maths of Game Theory* <https://www.gresham.ac.uk/watch-now/game-theory>.
- The rules of the Game of Ur were rediscovered by Irving Finkel, a researcher at the British Museum, and there's a brilliant [YouTube video](#) of him explaining them and playing a game with Tom Scott.
- <https://BoardGameGeek.com> is an extensive website that is the home for a large community of board game fans, with information and discussion about a huge range of games.
- Play Conway's Game of Life at <https://playgameoflife.com/>
- If you want to know exactly how many legal Connect Four positions there are after  $n$  moves, there's of course a website that tells you. Check it out here <https://oeis.org/A212693>
- Financial Times article "[Man beats machine at Go in human victory over AI](#)" 17 February 2023.
- Shannon's article on computer chess is available from the Computer History Museum website: <https://www.computerhistory.org/chess/doc-431614f453dde/>
- The British Go Association website has the rules of Go and much more: <https://www.britgo.org/>

## PowerPoint Presentation Image Credits

To the best of my knowledge, all images not mentioned below are either created by me, are in the public domain, or may be used for educational purposes under fair use rules.

- [Ur scratched on 8<sup>th</sup> century BCE statue](#) by Jack1956, [CC BY-SA 3.0](#), via Wikimedia Commons.
- [Pachisi being played in Tamil Nadu with tamarind seeds and stones](#), by Surya Prakash.S.A., [CC BY-SA 3.0](#), via Wikimedia Commons.
- [https://commons.wikimedia.org/wiki/File:Nine\\_Men%27s\\_Morris\\_with\\_dice\\_in\\_Libro\\_de\\_los\\_juegos.jpg](https://commons.wikimedia.org/wiki/File:Nine_Men%27s_Morris_with_dice_in_Libro_de_los_juegos.jpg)
- Campaign for North Africa pictures by Charles Picard, at <https://boardgamegeek.com/boardgame/4815/campaign-north-africa-desert-war-1940-43>
- [Ludo board](#) by Micha L. Rieser, via Wikimedia Commons
- [Mancala variant called kala](#) by Matěj Bařha, [CC BY-SA 2.5](#), via Wikimedia Commons
- [Solitaire board](#) by Annielogue, [CC BY 3.0](#), via Wikimedia Commons
- Game of life "[pulsar oscillator](#)" gif by JokeySmurf and "[glider](#)" gif by Rodrigo Silveira Camargo both public domain, via Wikimedia Commons; [Bill Gosper's Glider Gun](#) image by [Lucas Vieira](#), [CC BY-SA 3.0](#), via Wikimedia Commons, made using Life32 v2.15 beta, by Johan G. Bontes.
- [Chomp game](#), by Lord Belbury, [CC BY-SA 4.0](#), via Wikimedia Commons.
- [Photograph of Claude Shannon](#) by Jacobs, Konrad, [CC BY-SA 2.0 DE](#), via Wikimedia Commons.
- Go examples courtesy of the [British Go Association](#) at this page <https://www.britgo.org/intro/intro2.html>.

© Professor Sarah Hart 2023