

Ultimate tic-tac-toe

Ultimate tic-tac-toe (also known as **super tic-tac-toe**, **strategic tic-tac-toe**, **meta tic-tac-toe**, **tic-tac-tic-tac-toe**, or **(tic-tac-toe)**^[1]) is a board game composed of nine tic-tac-toe boards arranged in a 3 × 3 grid.^{[2][3]} Players take turns playing in the smaller tic-tac-toe boards until one of them wins in the larger tic-tac-toe board. Compared to traditional tic-tac-toe, strategy in this game is conceptually more difficult and has proven more challenging for computers.^[4]

Contents

Rules

Gameplay

Computer implementations

References

External links

Rules

Each small 3 × 3 tic-tac-toe board is referred to as a local board, and the larger 3 × 3 board is referred to as the global board.

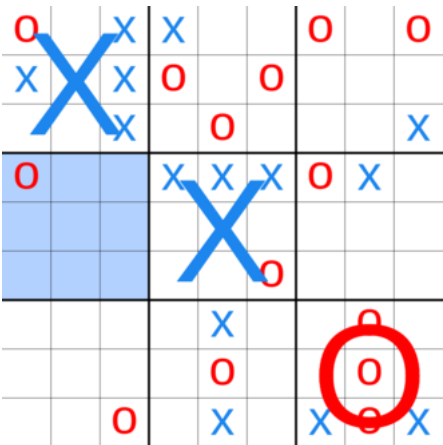
The game starts with X playing wherever they want in any of the 81 empty spots. This move "sends" their opponent to its relative location. For example, if X played in the top right square of their local board, then O needs to play next in the local board at the top right of the global board. O can then play in any one of the nine available spots in that local board, each move sending X to a different local board.

If a move is played so that it is to win a local board by the rules of normal tic-tac-toe, then the entire local board is marked as a victory for the player in the global board.

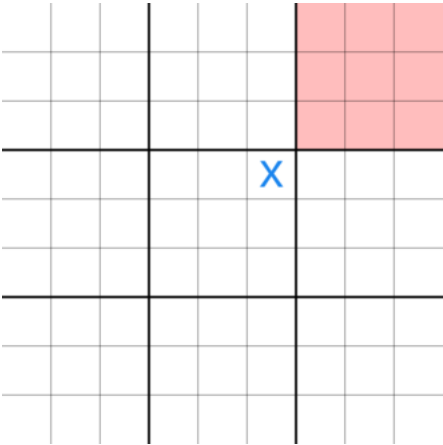
Once a local board is won by a player or it is filled completely, no more moves may be played in that board. If a player is sent to such a board, then that player may play in any other board.

Game play ends when either a player wins the global board or there are no legal moves remaining, in which case the game is a draw.^[3]

Gameplay



The board of an incomplete ultimate tic-tac-toe game (the large "X"s and "O"s representing local board games that have been won by that player). The most recent move was O playing in the middle-left square of the top-middle grid, forcing X to play their next move in the middle-left grid.



Since X played in the top-right corner of the local board, O is forced to play their next move in the top-right local board.

Ultimate tic-tac-toe is significantly more complex than most other variations of tic-tac-toe, as there is no clear strategy to playing. This is because of the complicated game branching in this game. Even though every move must be played in a local board, equivalent to a normal tic-tac-toe board, each move must take into account the global board in several ways:

1. **Anticipating the next move:** Each move played in a local board determines where the opponent's next move may be played. This might make moves that may be considered bad in normal tic-tac-toe viable, since the opponent is sent to another local board, and may be unable to immediately respond to them. Therefore, players are forced to consider the larger game board instead of simply focusing on the local board.
2. **Visualizing the game tree:** Visualizing future branches of the game tree is more difficult than single board tic-tac-toe. Each move determines the next move, and therefore reading ahead—predicting future moves—follows a much less linear path. Future board positions are no longer interchangeable, each move leading to starkly different possible future positions. This makes the game tree difficult to visualize, possibly leaving many possible paths overlooked.
3. **Winning the game:** Due to the rules of ultimate tic-tac-toe, the global board is never directly affected. It is only governed by actions that occur in local boards. This means that each local move played is not intended to win the local board, but to win the global board. Local wins are not valuable if they cannot be used to win the global board—in fact, it may be strategic to sacrifice a local board to your opponent in order to win a more important local board yourself. This added layer of complexity makes it harder for humans to analyze the relative importance and significance of moves, and consequently harder to play well.

Computer implementations

While tic-tac-toe is elementary to solve^[5] and can be done nearly instantly using depth-first search, ultimate tic-tac-toe cannot be reasonably solved using any brute-force tactics. Therefore, more creative computer implementations are necessary in order to play this game.

The most common artificial intelligence (AI) tactic, minimax, may be used to play ultimate tic-tac-toe, but has difficulty playing this. This is because, despite having relatively simple rules, ultimate tic-tac-toe lacks any simple heuristic evaluation function. This function is necessary in minimax, for it determines how good a specific position is. Although elementary evaluation functions can be made for ultimate tic-tac-toe by taking into account the number of local victories, these largely overlook positional advantage that is much harder to quantify. Without any efficient evaluation function, most typical computer implementations are weak, and therefore there are few computer opponents that can consistently outplay humans.^[4]

However, artificial intelligence algorithms that don't need evaluation functions, like the Monte Carlo tree-search algorithm, have no problem in playing this game. The Monte Carlo tree search relies on random simulations of games in order to determine how good a position is instead of a positional evaluation and is therefore able to accurately assess how good a current position is. Therefore, computer implementations using these algorithms tend to outperform minimax solutions and can consistently beat human opponents.^{[2][6]}

References

1. Epstein, Dave. "NP Completeness in Contemporary Board Games" (<http://www.columbia.edu/~ne2260/theory/games.html>).
2. Whitney, George; Janoski, Janine (November 26, 2016). "Group Actions on Winning Games of Super Tic-Tac-Toe". arXiv:1606.04779 (<https://arxiv.org/abs/1606.04779>). Bibcode:2016arXiv160604779G (<https://ui.adsabs.harvard.edu/abs/2016arXiv160604779G>).
3. Orlin, Ben (2013-06-16). "Ultimate Tic-Tac-Toe" (<https://mathwithbaddrawings.com/2013/06/16/ultimate-tic-tac-toe/>). *Math with Bad Drawings*. Retrieved 2016-10-18.

4. Lifshitz, Eytan; Tsurel, David (December 26, 2016). "AI Approaches to Ultimate Tic-Tac-Toe" (<http://www.cs.huji.ac.il/~ai/projects/2013/UltimateTic-Tac-Toe/files/report.pdf>) (PDF). *The Rachel and Selim Benin School of Computer Science and Engineering*.
5. Schaefer, Steve (2002). "MathRec Solutions (Tic-Tac-Toe)" (<http://www.mathrec.org/old/2002jan/solutions.html>). Retrieved 2016-10-18.
6. Gila, Ofek (2016-06-27). "What is the Monte Carlo tree search?" (<http://blog.theofekfoundation.org/artificial-intelligence/2016/06/27/what-is-the-monte-carlo-tree-search/>). *We Blog*. Retrieved 2016-10-18.

External links

- [Monte Carlo tree-search implementation](http://www.theofekfoundation.org/games/UltimateTicTacToe/) (<http://www.theofekfoundation.org/games/UltimateTicTacToe/>)
 - [Open Source Monte Carlo tree search implementation in C and Swift](https://github.com/mkhrapov/tictactoe-ultimatum/) (<https://github.com/mkhrapov/tictactoe-ultimatum/>)
 - [Ultimate Tic-Tac-Toe game where artificial intelligences confront each other](https://www.codingame.com/multiplayer/bot-programming/tic-tac-toe) (<https://www.codingame.com/multiplayer/bot-programming/tic-tac-toe>)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Ultimate_tic-tac-toe&oldid=948438751"

This page was last edited on 1 April 2020, at 01:32 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.