Puzzle Generation for Ultimate-Tic-Tac-Toe

Maj Zirkelbach
mz5153@student.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Aleksander Sadikov
aleksander.sadikov@fri.uni-lj.si
University of Ljubljana, Faculty of Computer and Information Science
Ljubljana, Slovenia

Abstract

Ultimate Tic-Tac-Toe is an interesting and popular variant of Tic-Tac-Toe that lacks available resources for improving gameplay skills. In this paper, we present a semi-automatic system for generating puzzles as a part of a larger tutorial application aimed at teaching Ultimate Tic-Tac-Toe. The puzzles are designed to enhance players' tactical and strategic understanding by presenting game scenarios where they must identify correct continuations. To ensure the quality of generated puzzles we tested the application with a group of volunteers. The results have shown that the number of solved puzzles positively impacted users' ability to reach higher strength levels but had less of an effect on lower levels.

Keywords

Ultimate Tic-Tac-Toe, puzzle generation, minimax algorithm, tutor application

1 Introduction

For centuries, people have enjoyed playing board games like chess and Go. Over time, these games have led to the development of extensive theory and the accumulation of knowledge, helping players navigate their complexity. Today, advanced artificial intelligence (AI) programs such as AlphaZero [14] surpass even the strongest human players, offering new insights into strategies. However, many lesser-known games have yet to be thoroughly explored, despite their popularity. One such game is Ultimate Tic-Tac-Toe, an advanced version of the classic Tic-Tac-Toe. This game is played on a 3x3 grid of local Tic-Tac-Toe boards, creating a global board (Figure 1a). The goal is to win three local boards in a row, while players must make their moves within specific local boards determined by their opponent's previous move. For example, if a player moves in the top-left corner of a local board, the next player must play on the top-left local board. If the designated board is full or decided, the player can choose any other available board. Despite its apparent simplicity, the game has enough spatial complexity that it cannot currently be solved using brute-force methods.

While there are several online implementations of the game, most focus on building strong AI agents; however, There is a noticeable lack of resources aimed at teaching and helping players understand the deeper strategies of the game, which could make the learning curve more manageable for new and aspiring players. Thus, we have created an application that addresses the lack of learning tools available for Ultimate Tic-Tac-Toe. This article places particular emphasis on the puzzle generation aspect of our application, which is designed to enhance players' tactical and strategic thinking.

In Section 2 we present the related work, and in Section 3 we detail the technical aspects of the developed application. In Section 4 we present the implemented agents and their approximate strength. In Section 5 we provide a description of different types of puzzles and the methodology for their construction. In Section 6 we present the evaluation and discuss the results in Section 7. Finally, in Section 8 we present the conclusions and give possible extensions and enhancements for future work.

2 Related Work

There are many implementations of the Ultimate Tic-Tac-Toe available online, mostly appearing as mobile games aimed primarily at entertainment and lacking advanced playing agents [12] [9] [10], as well as web and desktop applications developed to create the strongest possible programs [15] [7] [13]. An example of the latter is an agent based on the ideas of the AlphaZero program [14], currently considered one of the strongest players of this game [13]. During the development of this agent, significant strategies were discovered, which were also useful in developing our application. Some researchers have attempted to solve the game theoretically, but the spatial complexity proved too great to allow for a complete solution [5].

It is important to differentiate between the various versions of Ultimate Tic-Tac-Toe. One variant allows the game to continue playing on already-won local boards, which drastically changes the game's dynamics. In this variant, researchers have demonstrated an optimal strategy for the starting player, who can win in 43 moves [1]. Further research has focused on enabling a more balanced game by introducing random opening moves, which reduces the predictability of forced wins [4]. Despite these interesting findings, research on these variants is not so relevant for us, as it does not contribute to the understanding of the main game.

While there is a lack of educational material specific to our game, much can be learned from related fields, such as chess, which has been extensively researched. The paper by Gobet and Jansen [8] describes a scientific approach to learning chess, which includes methods to improve memory, perception, and problem-solving skills in players. In this context, it focuses on the acquisition and organization of knowledge, including both explicit and implicit learning of tactics and strategies. This approach facilitates a deep understanding of games and the development of more effective learning methods.

Chess also offers highly sophisticated practical solutions from which we can learn a great deal. Platforms such as `chess.com` [2] and `lichess.org` [11] offer extensive resources and tools for learning chess, especially in the areas of tactics and openings. These platforms allow players to learn through interactive lessons, solving puzzles, and studying various openings, which contribute to a deeper understanding of the game and improve playing skills.

Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia

https://doi.org/10.70314/is.2024.scai.7299

This approach has proven extremely effective in helping players master complex strategic and tactical concepts in chess.

On the mentioned platforms, the methods for learning tactics are designed to allow players to solve problems based on concrete game situations, which improves pattern recognition and decision-making abilities in real games. Similarly, learning openings involves demonstrating optimal opening moves and their continuations, helping players develop effective strategies at the beginning of the game.

We have applied similar methods in our Ultimate Tic-TacToe application. For example, adapting approaches for learning tactics can help users improve their recognition and solving of complex situations in the game while learning openings helps to understand key opening moves and their impact on the further course of the game. By incorporating these methods into our application, we ensured more effective learning processes and improved the overall gaming experience.

3 Application Details

In addition to puzzle-solving, the app offers a comprehensive learning experience through various other features. It includes AI opponents of different difficulty levels, game analysis, and exploration of effective opening strategies, allowing players to refine their understanding in all phases of the game. The user interface ensures smooth navigation between these modes, making the app a versatile tool for both playing and learning Ultimate Tic-Tac-Toe. By integrating these elements, the app serves as a resource for players at all levels, helping them to deepen their understanding and improve their skills.

To reach a broader audience, the application was developed for both Android and Windows, the dominant operating systems in the market [15]. It uses Flutter components to deliver a responsive and user-friendly interface. Local data storage is utilized for user settings, progress, and puzzle data, ensuring efficient performance and data management.

We employed modern technologies and mobile development practices, including state management patterns, to create an easily expandable app for future updates and enhancements. The entire project is hosted on GitHub, though it is not open-source.

Test versions of the app for Android and Windows are available on Google Drive: https://drive.google.com/drive/folders/1SnO_mN_ZVa2wXd0OGI07kLiYKQTDHuEe?usp=drive_link, while the Android production version is accessible on Google Play Store: https://play.google.com/store/apps/details?id=com.uttt_tutor.

4 AI Agents and Rating System

Playing against intelligent agents allows users to refine their skills by competing against various virtual opponents. The application includes nine different agents, each varying in difficulty and gameplay strategies. These agents are designed using Minimax and Monte Carlo Tree Search [3] algorithms, which provide different levels of complexity and depth in move analysis. The agents and their approximate strengths are shown in Table 1.

To better understand the quality of the agents and evaluate user progress, we need to establish a system for measuring their strength. Since Ultimate Tic-Tac-Toe is not widely popular, there is no established system for rating player abilities. Therefore, we decided to use the chess rating system as an approximation for our agents.

The chess rating system is used to measure the playing strength of chess players. The most commonly used system is the Elo rating [6], which predicts the likelihood of one player winning against another based on their ratings:

$$E_A = \frac{1}{1 + 10^{\frac{R_B - R_A}{400}}},$$

where E_A represents the expected score for player A, R_A is the rating of player A, and R_B is the rating of player B.

Table 1: Table of approximate agent strengths. Each agent played 100 games (50 as X and 50 as O) against the agent one level lower. The results column shows the number of points each agent earned with each symbol, as well as the total score. A win awarded 1 point, while a draw awarded 0.5 points. The last line shows the results of the strongest freely available agent against level 9. It had the same amount of time to think, and they played 30 games.

Agent	Result			Estimated Rating
	X	O	Combined	
Confused Chimp - 1	-	-	-	1
Goofy Goblin - 2	49	49	98	620
Casual Carl - 3	41.5	35.5	77	835
Average Joe - 4	37	25	62	926
Hustling Hugo - 5	39.5	34.5	74	1114
Witty Walter - 6	43	30	73	1293
Thinking Tiffany - 7	35	24	59	1361
Brainy Bob - 8	42,.5	26.5	69	1506
Bossman - 9	36.5	22.5	59	1574
UTTT AI	14.5	12.5	27	1948

5 Puzzle Description and Methodology

In this section, we describe different types of puzzles and the methodology employed to generate them for our game.

5.1 Puzzles

The puzzles in the application are divided into tactical and strategic, with each type of puzzle covering different aspects of the game and helping players improve specific skills.

Tactical puzzles are useful for understanding tactical ideas and are particularly applicable in the endgame and middlegame phases. They focus on specific situations that require precise and thoughtful moves, helping players develop the ability to think quickly and effectively. In total, we generated 1,263 tactical puzzles, distributed across five levels. The number of puzzles for each level is shown in Table 2.

Unlike tactical puzzles, strategic puzzles aim to understand the position and long-term plans. They are instrumental in the opening and middlegame, where it is crucial to recognize strategic ideas and develop plans that provide an advantage as the game continues. There are 50 strategic puzzles available, currently arranged in one level, with the possibility of expansion in the future.

5.2 Tactical Puzzle Generation

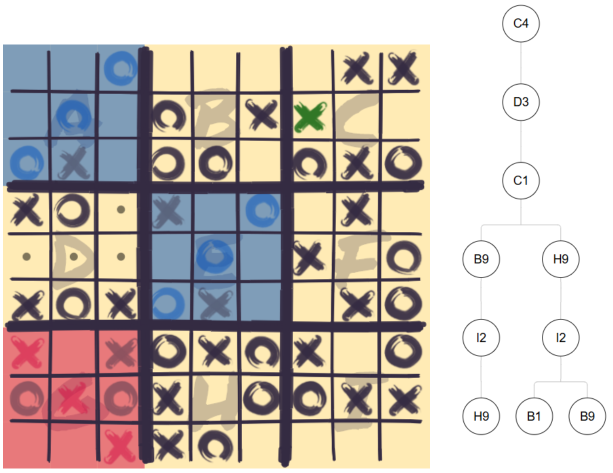
To generate tactical puzzles, we developed a specialized minimax agent that builds a tree of all possible moves leading to victory from the solver's perspective. A key step in this process is the

Table 2: Number of tactical puzzles on each level.

Level	Puzzle Depth	Quantity
1	1	273
2	3	493
3	5	231
4	7	176
5	9	90

selection of tree branches to retain only relevant and correct solutions. It is essential to preserve all of the winner's possibilities while limiting the loser's responses to those that make finding a solution as difficult as possible. Therefore, we select the continuation that allows the longest possible game for the loser while leading to the fewest continuations for the winner.

From the tree, we extract all the correct solutions for the given position. For a high-quality puzzle, it must not have too many solutions. The criterion we set is that the number of solutions must be less than the depth of the puzzle. We also decided to discard all puzzles that have multiple correct continuations for the first move. This way, we avoid trivial puzzles that would be too simple. An example of a level 3 tactical puzzle with its generated solution tree is shown in Figure 1.



(a) Level 3 tactical puzzle. **(b) Solution tree.**

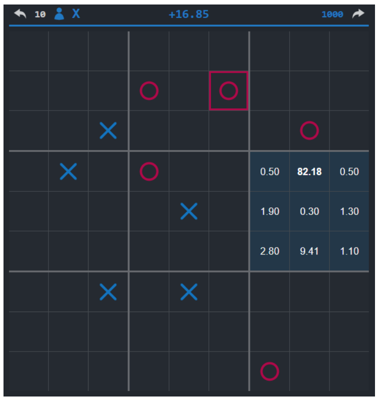
Figure 1: An example of tactical puzzle and its generated solution tree.

The generation of tactical puzzles for different difficulty levels was automated by conducting matches between agents of equal strength, with the search depth of both agents corresponding to the depth of the puzzle we wanted to find. We chose this approach to ensure that the resulting positions were interesting and balanced, as otherwise, the stronger side would usually have an overly obvious advantage at the start of the puzzle which would make it boring to solve.

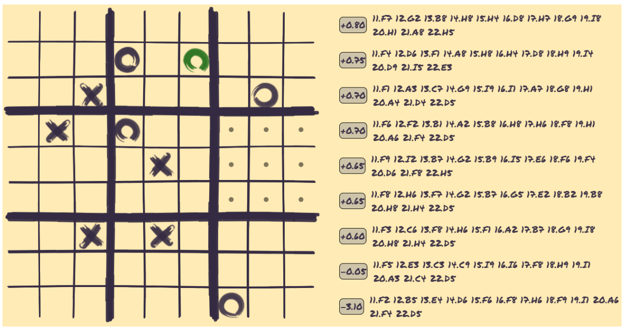
5.3 Strategic Puzzle Generation

Automating the creation of strategic puzzles is impossible without a program that could interpret the given position and simultaneously provide a human-understandable explanation. Additionally, generating strategic puzzles requires an agent with an advanced strategic understanding of the position, which our agents, using relatively simple heuristics, are incapable of. Therefore, we resorted to the most powerful freely available agent [13], which is based on the ideas of the AlphaZero program.

Thus, we generated the strategic puzzles manually. We searched for interesting and instructive positions that arose in games between the aforementioned agent and our stronger programs. We focused on moments when there was a significant deviation in the position evaluation between the two agents. When the agent with better strategic understanding detected an important change, we saved the given position, studied it more closely, and based on our understanding of the game, formulated a solution. The most common examples of such situations involved sacrificing the edge board to gain control over the central board. A basic example of this can be seen in Figure 2.



(a) User interface of the most powerful freely available agent. For the given position, it ran 1000 simulations and assessed the move F2 as the best with an 82% probability. It evaluates the position with a value of +16.85, which means it assigns approximately 58.4% win probability to player X (a value of 0 means a draw, 100 a win, and -100 a loss).



(b) Minimax agent with a search depth of 12. It marks the move F2 as the worst, as it does not recognize the long-term advantage.

Figure 2: Different interpretations of the same position, based on which we built the strategic puzzle.

6 Evaluation and Results

We conducted a quality analysis of the application with 14 volunteers. Their task was to use the app for an extended period to improve their knowledge of the game. We were interested in determining whether using the app had a positive impact on the development of their Ultimate Tic-Tac-Toe playing skills and whether progress was dependent on motivation or the time spent learning.

To assess individual progress, participants played against the agent at the start of testing to determine their initial skill level. The application then tracked the highest level each user defeated, providing an estimate of their improvement over time. This progress, in relation to the number of solved puzzles solved, is illustrated in Figure 3. For a more concrete interpretation of obtained level strengths, refer to Section 4.

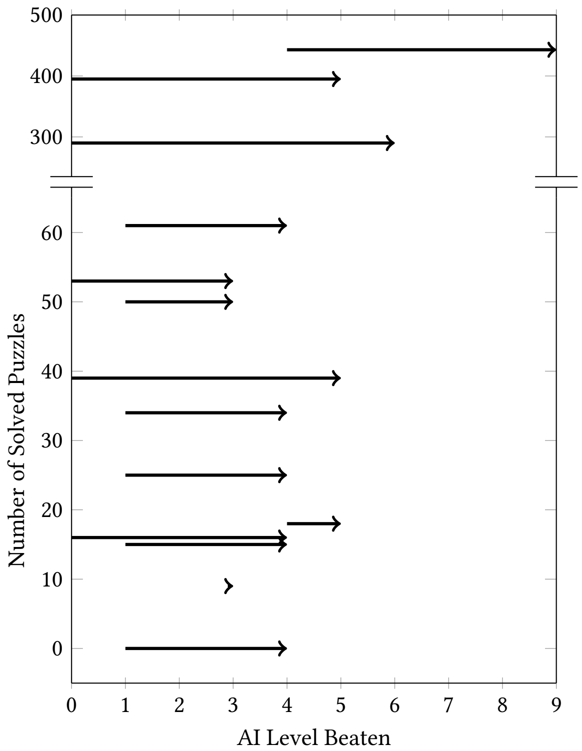


Figure 3: Progress in relation to the number of solved puzzles. Each arrow represents a human tester and indicates the change in the achieved level from the beginning to the end of the application's use.

7 Discussion

The results in Figure 3 indicate that solving more puzzles impacted users' ability to reach higher levels, but had less of an effect on lower levels. This is likely due to the fact that beginners can improve relatively quickly by simply playing the game, whereas advanced players require more effort to progress (eg. it is a lot easier to gain 100 rating points when you are rated 500 as compared to when you are rated 1500).

The reason for this is that at lower ratings, there is generally more room for rapid improvement because the skill gap between players tends to be more pronounced, and beginners can quickly benefit from fundamental knowledge and tactical awareness. As a result, achieving a higher rating initially is easier as players can fix obvious mistakes and exploit weaker opponents' errors.

However, as players reach higher levels, the competition becomes tougher, and the differences in skill become more nuanced. Players at this level are more consistent and less likely to make blunders, so improving further requires mastering advanced strategies, pattern recognition, and deeper positional understanding, making progress slower and more challenging. This reflects the diminishing returns on improvement as you climb the rating ladder.

It must also be mentioned that users were free to use any tools within the app during testing and solving more puzzles did not correlate with longer app usage. For a clearer assessment of puzzle significance, a controlled test focusing solely on puzzle-solving would be more appropriate.

8 Conclusion

In this work, we presented methods for generating puzzles for the game of Ultimate Tic-Tac-Toe. To evaluate the quality of these puzzles, we tracked how the number of solved puzzles impacted individual user progress. Our results indicate a correlation between the number of puzzles solved and the ability to reach stronger AI levels.

However, the evaluation could be refined by focusing exclusively on the puzzle-solving component, isolating it from other functionalities of the application. Additionally, the automation of tactical puzzle generation could be expanded to cover the middlegame phase, rather than being limited to endgame scenarios. Another area of improvement is providing clearer assessments of puzzle difficulty. This could be achieved by implementing a rating system that ranks puzzles based on completion rates, offering a more accurate measure of challenge for each puzzle.

Acknowledgements

The author would like to thank the family and friends who participated in testing the application.

References

[1] Guillaume Bertholon, Rémi Géraud-Stewart, Axel Kugelmann, Théo Lenoir, and David Naccache. 2020. At most 43 moves, at least 29: optimal strategies and bounds for ultimate tic-tac-toe. (2020). DOI: 10.48550/ARXIV.2006.02353.
[2] Chess.com. 2024. Chess.com. (June 2024). https://www.chess.com/.
[3] Rémi Coulom. 2006. Efficient selectivity and backup operators in monte-carlo tree search. In *In: Proceedings Computers and Games 2006*. Springer-Verlag.
[4] Justin Diamond. 2022. A practical method for preventing forced wins in ultimate tic-tac-toe. (2022). DOI: 10.48550/ARXIV.2207.06239.
[5] Nelson Elhage. 2020. Solving ultimate tic tac toe. (July 2020). https://minimax.dev/docs/ultimate/.
[6] Arpad E Elo. 1978. *The Rating of Chessplayers, Past and Present.* Arco Pub.
[7] Ofek Gila. 2019. Ultimate tic tac toe. (2019). https://www.theofekfoundation.org/games/UltimateTicTacToe/.
[8] Fernand Gobet and Peter J Jansen. 2006. Training in chess: a scientific approach. *Education and chess.*
[9] Henryk. 2023. Ultimate tic tac toe. (Dec. 2023). https://play.google.com/store/apps/details?id=com.henrykvdb.sttt.
[10] HPStudios. 2024. Ultimate tic tac toe. (June 2024). https://play.google.com/store/apps/details?id=com.MertTaylan.UltimateTicTacToe.
[11] Lichess. 2024. Lichess. (June 2024). https://lichess.org/.
[12] Levi Moore. 2020. Ultimate tic tac toe. (Nov. 2020). https://play.google.com/store/apps/details?id=com.ZeroRare.UltimateTicTacToe.
[13] Arkadiusz Nowaczynski. 2021. Ar-nowaczynski/utttai: alphazero-like ai solution for playing ultimate tic-tac-toe in the browser. (Dec. 2021). https://github.com/ar-nowaczynski/utttai.
[14] David Silver et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362, 6419, 1140–1144. eprint: https://www.science.org/doi/pdf/10.1126/science.aar6404. DOI: 10.1126/science.aar6404.
[15] Michael Xing. 2019. Ultimate tic-tac-toe. (Oct. 2019). https://www.michaelxing.com/UltimateTTT/v3/.