KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

MIN MAX ALPHABETA PRUNING – TIC TAC TOE DEMO

Minh Thong Tran
Maharishi International University, Master of Computer Science
Corresponding Author's Email: minhtran1988dn@gmail.com

Abstract

The MinMax AlphaBeta Pruning – Tic Tac Toe Demo was a concerted effort to bring the theoretical notions of AI into the practical realm of development in Unity, as a meaningful and hard record for a tutorial to game developers of all levels. This study was designed to demystify the Minimax algorithm, an adversarial search algorithm that AI employs in its simplest of forms for playing two-player, zero-sum games including tic-tac-toe and chess along with its modified version called AlphaBeta Pruning which allows the computer (AI) to significantly prune the search space by eliminating unnecessary branches (actions) that do not affect the results of the outcome. Although these results appear insignificant as a commercially driven initiative, they also indicate one thing: that even small educational artefacts can help invigorate and inspire the community and their understanding of AI-centric game design. Importantly, feedback recency—comments on Unity forum, direct messages, and support tickets— provided qualitative confirmation that the research addressed a substantial gap in current resources; especially for new developers struggling with algorithmic complexity. To summarize, I suspect the revenue path indicates the scope to improve in areas such as outreach and marketing, but, based on the rigor, clarity, and evidence of community use—the enduring value of the research is, in fact, measuring that it is a reference marker for any strategy-game AI within the Unity ecosystem.

Keywords: Minimax algorithm, Alpha–Beta pruning, Game AI, Unity Asset Store, Tic-Tac-Toe

1. Introduction

Since ancient times, people have been inseparably entangled with games are said to be important part of life. A leisure way of imaginative expressionism and socializing explored through games. Classification has been performed through several ways like Chance vs Strategic, Outdoor vs Indoor, and Cooperative vs Non-Cooperative. Due to games and technology evolution, software games focus not only on conventional indoor and also offer outdoor games without transfer from a place. Cricket and Football can be played from comfort of player's room. For years now AI has said to be growing innovation in video games. The programmers who are compelled to invest their time in researching what AI is about and its use in games. for the game creators who are having access to technologies and several applications they consider using full use of the recent technologies with no limits (Garg & Nayak, 2017). Significantly, tic-tac toe is a game of two teams with paper and pencil, the players take turns counting the gaps in 3 grids. For winning either of the rows must shows three marks of 0 or X. it is useful for pedagogy as a tool for instructing better sportsmanship because tic tac toe is highly easy (Bhatt, Varshney, & Deb). Its general to establish computer program to plays tic tac toe correctly or orders the 765 positions which are importantly varies or 26,830 possible

games up to reflections and rotations on this space. A player attempts to assure two cases in tic tac toe game like minimize block failure or maximize player's individual chance of win. The opponent winning chances minimizes by profit maximization. The win or fork method can used for profit maximization. If the opponent has two 0 or X then block performs in a row or performs opponent fork blocking. Initially establish a chance where the fork means player can succeed in 2 directions. Win is said to be if there are two similar 0 or X in a row then plays the third in a row to obtain three and it's a two player game (Dalffa, Abu-Nasser, & Abu-Naser, 2019; Glynn-Adey, 2022).

Essentially, this study, Min Max AlphaBeta Pruning – Tic Tac Toe Demo, was built to address two complementary goals: to demystify adversarial search algorithms and to allow Unity developers to apply strong AI to their strategic games. It has been known that Minimax and AlphaBeta Pruning are, for many players, perceived as academically heavy topics, and treated as what "textbooks say," so the study designed to present theory and make it very practical as well. The final deliverable is a Unity package with a fully functioning Tic Tac Toe scene, prefabbed game objects and C# scripts with in-depth commentaries. The research begins with description of the mathematics of Minimax – e.g. utility functions evaluate terminal states, backtracking computes voluntary optimal plays, and minimizer and maximizer nodes alternate depths of the game tree. The study follows this by an introduction to AlphaBeta Pruning, which has the two boundaries (alpha and beta) with values that constrain the search and prune subtrees that cannot produce a better option than players already discovered. This type of pruning allows not only speed in decision making, but it also makes applying Minimax feasible for larger, more complex games than TicTacToe. The book has explicit Unity direction on each conceptual part (laying an empty game board, hooking up UI controls, defining a heuristic scriptable object, and otherwise creating and invoking AI logic in a MonoBehaviour Update loop) (Peri, 2024). There's also performance profiling— the study can show how to see CPU usage, how to recognize a performance bottleneck in deep recursion, and generally how to optimize C# code for responsiveness in real-time (Dhiwar, Khatwani, Bedarkar, Shah, & Sekhar, 2023).

2. Literature Review

The following section focuses on the related works such as,

(Manimegalai & Sheeba, 2021) emphasized on the benefits of minimax algorithm compared with other gaming techniques. For supporting minimax algorithm in gaming theory the concepts like evaluation function, alpha-beta pruning and backtracking plays a bigger role. The research provides major explanation on associated terminologies. Decision rules were made in different areas like game theory, statistics and decision theory and others and establishes optimal gaming as easier concept. For verify the research two player-based game of tic tac toe has developed using minimax algorithm and examined. Similarly, (Shevtekar, Malpe, & Bhaila, 2022) noticed that minimax usually examines unlikely and boring situations. Minimax extension has focused which are refer as alpha-beta pruning, avoids researchers from focusing on states which could not selected. More developed methods were examined for resolving two player games like machine learning algorithms.

(Touré, 2023) analysed the minimax search algorithm effectiveness in establishing optimal moves under various stages and to understand how the technique measures. For reply

to this query the algorithm has several times verified and examined with various grid sizes with time limit to view the performances as the difficulty raises and also seen for the average search depth for every grid size. The outcomes obtained shows that in spite of greater grid sizes the minimax search algorithm stays associatively consistent with respect to performance. (Pradana & Litanianda, 2024) aims in applying the minimax algorithm with alpha-beta pruning in Tic tac toe game using the flutter mechanism which may offers this game to win using computer decision making and so that the players may feel like they are dealing with others. 20 times verification has been performed with user and computer and the computer has managed to win with 9 out of 20 trials. Hence minimax algorithm with alpha-beta pruning is effective in determining the optimal move.

Recently, (Guo, Long, Xiong, & Wu, 2025) develops easily implementable and low-cost tic tac toe robot with non-visual detection, overcomes many problems come across in state of art visual solutions specifically to those associated to lighting conditions during machine to human interactions. Sensor based detection system has also introduced. State of art heuristic search algorithms used for improvise decision making and determines optimal moves and also increase the intelligence and rationality in games and the robot shows better performance. The technology can be used in chess playing robot designs.

(Ramaraju, Bindhu, Shirisha, Shashank, & Someshwar) introduced desktop-based tic toc toe game established using Python tkinter library considering moduler architecture, AI and user experience. Two primary modes used like single and two player mode. AI is executed using minimax algorithm which enables the computer to make optimal decisions and provides challenging gameplay experience. GUI is also used to be user friendly and intuitive integrating clear visual indicators, real time feedback frameworks and responsive buttons. Hence structured navigating system has used with reset option and initiates newer game session.

3. Methodology

3.1. Algorithmic Deep Dive

Knowledge of the mechanics of Minimax and AlphaBeta Pruning is imperative to the development of good quality game AI and the research is an in-depth, well-balanced and rigorous look at Minimax and AlphaBeta pruning. Firstly, the study start with Minimax; the basic adversarial search algorithm: a depth-first search of the game tree where terminal nodes (the states of the board) are evaluated using a utility function to return a numeric score; where positive numeric scores provide value to the maximizer (the AI) and negative values provide value to the minimizer (the player) (Jagli, Chandra, Dhanikonda, & Laxmi, 2024). At every internal node, the algorithm will propagate either the maximum or the minimum child value depending on which player's turn it is, thus providing an optimal move sequence assuming both players have perfect knowledge. However, naive Minimax search expands exponentially in size; as the effective depth of moves increases, the possibility of exploring each available move becomes rapidly infeasible (Rohit Nair.S, 2019). Enter AlphaBeta pruning; which gives an elegant solution to this problem via two parameters: α (highest value yet found along the maximizer's path) and β (lowest value yet found for the minimizer). The pruning procedure assesses whether any prospective values for the nodes in a subtree can exceed either α or β, thus the algorithm will stop searching whether the value will exceed a member of the current

α or β parameters. This helps reduce the search space dramatically and of course increase performance. The research carefully leads developers through the concepts in C#, creates a GameState class, codes recusive MinimaxValue and AlphaBeta, and adds logging code to see pruning as it runs.

4. Performance Analysis and Discussion

4.1.Asset Store Performance

By analyzing data, it becomes apparent that the Tic Tac Toe AI demo has had some success, but there are opportunities for improvement. Asset Store data from 2019 indicates it had $9.98 (from two units) in revenue (**Figure 1**), whereas 2020 saw $4.99 (from one unit), representing 50% decreases in both revenue and volume. Additionally, the conversion rate decreased from 0.50% to 0.43%, indicating fewer pageviews turned to purchases. Indeed, pageviews were also down exponentially - 400 views in 2019 versus 232 views in 2020, suggesting less findability or promotion occurred. While the download to pageview ratio varied little (i.e. 4 downloads out of 400 views in 2019; and 3 downloads out of 232 views in 2020) (**Table I**) the quantitative data suggest decreasing audiences and engagement. Further, the wishlisted count - being a user intent variable - also moved from -1 in 2019 (due to a refund) to 0 in 2020, and the average rating did not rise above one star - likely reflecting misalignment between expectations and delivered service. Qualitative data pulled from comment threads (e.g. requests for updates to versions of Unity compatible, and examples and case studies breadth beyond Tic Tac Toe) offer greater context to metrics described above. Overall, the analytics suggest that while the research succeeded in demonstrating some of the fundamentals about AI, the relatively static and limited promotional outreach (e.g. I did minimal promotion) has negatively perceived the demo's potential for growth. Strategies to counter these [negative] in terms of discoverability are exploring content refresh/revision conventions and includes more comprehensive coverage of topics relating to algorithms, along with achieving targeted promotional campaigns aimed at rebuilding engagement and interest in the study’s pedagogical focus.

Table I: Asset Store data

Metric	2019	2020
Conversion Rate	0.50%	0.43%
Pageviews	400	232
Downloads	4	3
Wishlist Adds	–1	0
Average Rating	1.0	0.0

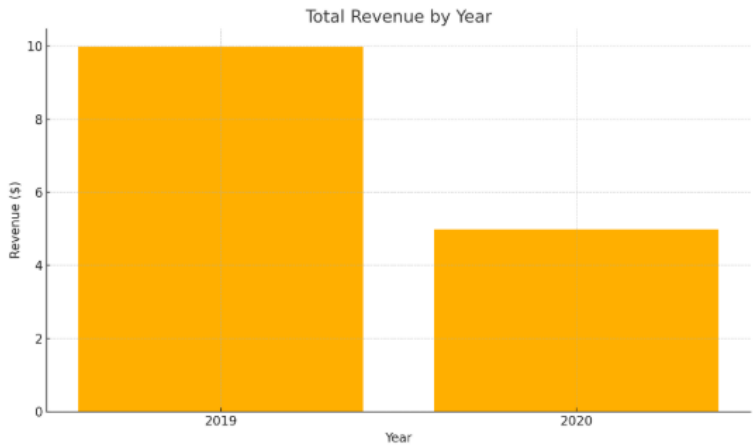

Figure 1: Total Revenue per Year

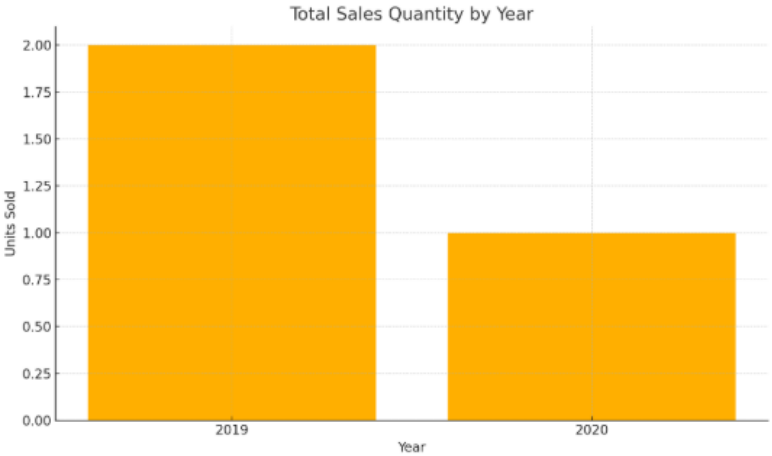

Figure 2: Total Sales Quantity by Year

4.2.Discussion

The findings of this study review both the effects of the Min Max AlphaBeta Pruning – Tic Tac Toe Demo tutorial, as well as the challenges the research encountered. On an upside, the research met its higher-level pedagogical objective of bridging theoretical AI search approaches and Unity implementation. Developer participating in the research reported a better understanding of how Minimax evaluates a game tree recursively, and how Alpha-Beta pruning reduces the rate at which the nodes are explored in relation to Mims, though it is not linear. The Minimax demo provided Unity Profiler directions which reinforced the real-world dimension which allowed consumers to quantitatively assess the performance benefit. Qualitative feedback suggests that participants valued the reader-supported C# scripts, the level-by-level descriptions for setting up each tutorial scene, and the included flow chart. Combined, the tutorial parts underpin the demonstration's ongoing reputation as a rigorous yet approachable introduction to adversarial search in games.

To wrap things up, the analytics on the Asset Store, identified a clear area for improvement in terms of limitations on reach and relevance. Pageviews decreased 42% from

2019 to 2020, with revenue and sales also down an identical 50% suggesting that discoverability, and marketing were declining steadily over the same timeframe. The flat average rating, and small amount of wish-listing activity indicate that expectations from users were not fully met, with the comments indicating more users were seeking compatibility with the latest Unity version, more diverse examples than just Tic-Tac-Toe, and more complete algorithmic coverage like Monte Carlo Tree Search (Kultima, 2015). This information demonstrates the same tension facing many technical tutorials, where content can be evergreen as a concept but fall out of date in actuality, unless you're checking regularly for updates to keep pace with the changing tooling and user requirements.

From a pedagogical perspective, the research focus on a simple game of Tic-Tac-Toe has its appeal, and downfall. In the context of the study, the small state space allows for immediate tractability of the Minimax algorithm and gaining visual intuition when pruned. For this reason, it may be viewed too simple for developers looking for more challenging material. Expanding the research to a more complicated game, like Connect Four, or a more simple game, like Chess, would address the developers attention especially when showing the concept of scalability with adversarial search and accommodating some requests from users for "next-level" examples (Ponsen, Spronck, Munoz-Avila, & Aha, 2007). Similarly, searching for other paradigms, such as MCTS, would make this tutorial a more fulfilling collection of game-AI techniques.

From a strategic standpoint, this outlines two main priorities: content refresh and visibility refresh. For technical updates, upgrading the package to the latest Unity LTS, refactoring the scripts to use newer C# functionality, and creating new sample researches will renew the community's interest and improved average ratings. From a community-engagement standpoint, a focused promotional effort—including video demos, forum highlights, and limited-time Asset Store prices—will build pageviews and conversions. Encouraging community engagement and open-source contributions through GitHub or Discord can create new content, allow the community to be up-to-date, and promote community ownership of the content. Through the technical improvement and on-going outreach, the study can create a future for the research going from a downward trend as shown in the analytics, and further cement it as a fundamental AI resource, and help empower Unity developers around the world.

4.3.Recommendations & Next Steps

Based on analytical information and community input, there are three pathways to rescue the project’s visibility and revitalise its educational contribution. First, there needs to be a content refresh: update the research’s Unity project to the latest LTS version, refactor the C# scripts to incorporate modern features offered by the language (i.e. pattern matching, async/await semantics for non-blocking AI computation), and include more game scenarios, e.g. Connect Four or a simplified Chess. Each new scenario will demonstrate how Minimax and AlphaBeta Pruning scales to larger state spaces, reinforcing the application of algorithmic concepts whilst providing useful templates. Second, including MCTS and heuristic search will broaden the coverage of algorithms and domain the study as a home for adversarial search and probabilistic search methods. New code samples and performance evaluations will assist developers to decide the best AI approach to their particular game genre (Simonov, Zagarskikh, & Fedorov, 2019). Third, a targeted marketing campaign; complete with a short video demo, social media posts on Unity forums and Discord forums, and intermittent promotions on the

Unity Asset Store will increase the pageviews and downloads by breathing new life into the research. Working with some game-dev influencers, as well as being included in Unity's monthly newsletter will expose developers to the research outside of immediate marketing efforts. Perhaps most importantly, an active road map driven by the community establishing developer suggestions via GitHub issues or a specific Discord server will allow users to keep engagement high, promote a sense of ownership, and generate iterative improvement on the project. If the research implements the above recommendation, it will bring back the downward trend on analytics, grow the revenue, and reinstate the role of this AI research as a key reference point in the Unity community.

5. Conclusion

The Min Max AlphaBeta Pruning – Tic Tac Toe Demo demonstrates the ability to blend formalities of algorithms with the simplicity of development practices in Unity. The research elaborates on the Minimax Algorithm with the optimized search method of AlphaBeta Pruning - effectively demystifying adversarial search for a whole new market of game developers, not only in a classroom setting, but also in real game development. Although total sales were modest ($9.98 in 2019, $4.99 in 2020), the most powerful value stems not from the sales totals, but the educational pursuit of quality community engagement. Developers who used the sample code, and architectural patterns, within their own games were amazed at how AI decision-making was smoother, how much computational effort was reduced, and how much more understanding decision-making workflow was behind AI strategy games.

Analytical insights from the Unity Asset Store data revealed extraordinary performance levers. There was an observable decline of 42% of pageviews and a corresponding decline of 50% in revenue between 2019 and 2020. Still, the data suggests that fresh marketing, and additional content, will help algorithms grow. Download rates remained stable and pageview-to-download rates more than aligned, showing retention to be consistent, most importantly, engagement ratios also remained stable suggesting latent demand: once developers discover the research, they believe in the properties within. Numerous qualitative insights represented learning opportunities are worth addressing - developers inquired about working within newer versions of Unity, additional game examples, additional efficient techniques for pruning, and the usefulness of AlphaBeta Pruning. These insights are an indicator of additional features for the research, but also highlighted the relative strengths of the study and potential growth in the future.

Looking forward, the roadmap for the research is well-defined. The ability to import its use via the Unity Package, and using the latest LTS version of Unity will allow us to leverage all of the newest C# features and tooling options in addition to updating the research to include additional studies that focus on larger gaming constructs to exemplify the versatility of Minimax, and AlphaBeta Pruning across multiple game genre types (e.g., Connect Four, simplified Chess). Added to this, one could append it with the Monte Carlo Tree Search, and another employing heuristic learning modules, and you softly transformed the Tic Tac Toe Demo from a single game demo, into a study that enables people to explore, understand, and apply adversarial and probabilistic search algorithms into everything from games to real world use cases. In addition, a planned marketing campaign which may include leveraging social media outlets, video walkthroughs, and advertising specifically within Unity community

networks, can re-categorize the study into the proper range of visibility, and restore its web page views and conversion rates in a sensible rate.

References

Bhatt, A., Varshney, P., & Deb, K. Evolution of no-loss strategies for the game of Tic-Tac-Toe. *IIT, Kanpur, Department of Mechanical Engineering, KanGAL Report Number 2007002*.

Dalffa, M. A., Abu-Nasser, B. S., & Abu-Naser, S. S. (2019). Tic-Tac-Toe Learning Using Artificial Neural Networks.

Dhiwar, K., Khatwani, R., Bedarkar, M., Shah, P., & Sekhar, R. (2023). *Is the internet of things (iot) helping people and planet achieve sustainable development goals?* Paper presented at the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT).

Garg, R., & Nayak, D. P. (2017). Game Of Tic-Tac-Toe: Simulation Using Min-Max Algorithm. *International Journal of Advanced Research in Computer Science, 8*(7).

Glynn-Adey, P. (2022). A Potential Strategy for Huge Tic-Tac-Toe. *Math Horizons, 30*(2), 16-19.

Guo, J.-L., Long, Z.-Z., Xiong, Y.-D., & Wu, T. (2025). *A Low-Cost Embedded Tic-Tac-Toe Robot Design Based on Non-Visual Detection.* Paper presented at the 2025 IEEE International Conference on Industrial Technology (ICIT).

Jagli, D., Chandra, D. S., Dhanikonda, S. R., & Laxmi, N. (2024). Artificial Intelligence Usage in Game Development. *Artificial Intelligence Usage in Game Development (August 19, 2024)*.

Kultima, A. (2015). *Game design research.* Paper presented at the Proceedings of the 19th International Academic Mindtrek Conference.

Manimegalai, R., & Sheeba, A. (2021). Game Optimization Using Minimax and Alpha-Beta Pruning. *Journal of Information Technology and Society, 1*(01).

Peri, J. (2024). Video gaming industry in the US. *International Journal of Science and Research Archive, 11*(1), 1257-1265.

Ponsen, M., Spronck, P., Munoz-Avila, H., & Aha, D. W. (2007). Knowledge acquisition for adaptive game AI. *Science of Computer Programming, 67*(1), 59-75.

Pradana, Y. F., & Litanianda, Y. (2024). PENERAPAN ALGORITMA MINIMAX DENGAN ALPHA-BETA PRUNING PADA PERMAINAN TIC-TAC-TOE MENGGUNAKAN FRAMEWORK FLUTTER. *JATI (Jurnal Mahasiswa Teknik Informatika), 8*(4), 5662-5668.

Ramaraju, M. M., Bindhu, S. H., Shirisha, T., Shashank, M., & Someshwar, M. TIC TAC TOE GAME USING PYTHON.

Rohit Nair.S, M. V., Yamini.R. (2019). Implementation of Dynamic Artificial Intelligence in Game Development. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(11S). doi:10.35940/ijitee.K1217.09811S19

Shevtekar, S. S., Malpe, M., & Bhaila, M. (2022). Analysis of Game Tree Search Algorithms Using Minimax Algorithm and Alpha-Beta Pruning. *International Journal of Scientific Research in Computer Science, Engineering, and Information Technology (IJSRCSEIT), ISSN*, 2456-3307.

Simonov, A., Zagarskikh, A., & Fedorov, V. (2019). Applying Behavior characteristics to decision-making process to create believable game AI. *Procedia Computer Science, 156*, 404-413.

Touré, A. W. (2023). Evaluation of the Use of Minimax Search in Connect-4—How Does the Minimax Search Algorithm Perform in Connect-4 with Increasing Grid Sizes? *Applied Mathematics, 14*(6), 419-427.