Project Report
2048 Game

Monte Carlo Tree Search and Games

Master IASD
Université PSL

Alejandro Jorba, Efstathios Chatziloizos

15/04/2025

1. Introduction

2048 is a popular single-player puzzle game that combines simple mechanics with complex strategic depth. Played on a 4×4 grid, the objective is to merge tiles with the same value by sliding them in one of four directions: up, down, left or right. After each move, a new tile of value 2 or 4 appears in a random empty cell. The game ends when no valid moves remain and the player's goal is to reach the *2048* tile or achieve the highest possible score.

Despite its apparent simplicity, 2048 poses a significant challenge due to its stochastic and partially observable nature. Each move introduces randomness in the form of new tile placements and while the rules are deterministic, optimal play requires long-term planning and adaptability to uncertain outcomes.

Monte Carlo Tree Search (MCTS) is a family of search algorithms widely used in decision-making problems where the environment is uncertain or complex. MCTS has achieved notable success in board games like Go and Chess, outclassing human players by a large margin. Its core strength lies in balancing exploration of new strategies with exploitation of known successful ones, making it well-suited to environments where evaluating the entire state space is computationally infeasible.

The motivation for applying MCTS to *2048* stems from its ability to plan ahead under uncertainty, a property essential for success in the game. Unlike traditional search methods or greedy heuristics that fail to account for randomness in tile generation, MCTS can simulate future game states, weigh the expected outcomes of different moves and iteratively refine its policy through sampling.

In this project, we explore and compare several MCTS variants on the *2048* game. By doing so, we aim to better understand how advanced search techniques can be adapted to complex stochastic environments and what insights they reveal about strategic planning in the game.

2. Methodology

To evaluate the effectiveness of Monte Carlo Tree Search on *2048*, we implemented and compared several MCTS variants: Classic Monte Carlo, Upper Confidence Bounds for Trees (UCT), Nested Monte Carlo Search (NMCS) and Nested Rollout Policy Adaptation (NRPA). All algorithms were evaluated in a fixed-horizon configuration, in which rollouts or simulations were truncated at a predefined depth. This setting reflects practical constraints in real-time environments and reduces the variance associated with unbounded rollouts, especially in a domain where terminal states can be many steps away.

The fixed-horizon design was uniformly applied across all methods to ensure comparability. We also explored full rollout variants in preliminary testing; however, the computational cost for certain methods was prohibitively high. This highlighted a key bottleneck in the current implementation of the game engine. Notably, performance could likely be improved by optimizing state representation. The existing engine uses a naïve 2D array to model the board, which introduces overhead in move generation and evaluation. A more efficient alternative would involve encoding board states using bitboards and leveraging precomputed lookup tables with bitwise operations for fast move execution. This would substantially reduce simulation time, particularly beneficial in approaches that involve large numbers of rollouts.

Both NRPA and NMCS rely on multi-level recursive simulations that increase the computational burden as the number of levels is raised. In our experiments, we found that deeper lookahead, controlled by both the simulation (rollout) depth and the number of nesting levels, led to an improvement in the quality of the chosen move and, consequently, the overall game performance. More specifically, we noticed performance improvements up until level 3. In fact, even a level 3 configuration with a very small lookahead outperformed a level 2 configuration that used a substantially larger rollout depth, for both NMCS and NRPA. However, this increased accuracy came with a substantial runtime cost. The recursive structure of these algorithms causes an exponential growth in the number of simulated playouts as the levels increase. In practice, NRPA and NMCS required considerably more time to run compared to classic Monte Carlo methods. To ensure fair comparability across methods, we adopted a fixed-horizon approach for the rollouts, thereby limiting each simulation to a predefined depth while systematically tuning the number of levels

3. Results

NMCS consistently outperformed other methods, frequently reaching the 2048 tile and achieving average scores around 23000. Inspection of verbose logs (by setting the verbose flag of the script to True) revealed that NMCS tended to favor action sequences that preserved high-value tiles in corner positions, which is a heuristic/strategy commonly used by human players. This behavior is particularly interesting given the fact that no domain-specific bias was explicitly encoded into the algorithm.

NRPA, despite its theoretical strengths, performed significantly worse than expected, yielding average scores of approximately 3000. The big difference likely stems from the interaction between policy adaptation and environmental stochasticity. Since NRPA's adaptation mechanism is sensitive to reward signals, the noisy and delayed feedback in *2048* most likely hindered effective policy shaping without additional regularization or value bootstrapping.

Classic Monte Carlo and UCT achieved comparable performance, with average scores in the range of 6000. The similarity between these two methods suggests that the stochasticity in the game dilutes the benefits of UCT's exploration term, particularly given the limited horizon and sparse rollout returns.

4. Conclusion

In summary, our exploration of Monte Carlo Tree Search variants for the 2048 game has shown the strengths and challenges of applying advanced, recursive search techniques to a stochastic, single-player environment. NMCS proved particularly effective, consistently achieving high scores (often reaching the 2048 tile) and demonstrating human-like strategies such as preserving high-value tiles in corner positions, even without any explicit domain-specific bias. In contrast, while NRPA theoretically offers the ability to refine a policy through recursive adaptation, its performance in our experiments was noticeably lower. We attribute this discrepancy to the sensitivity of NRPA's adaptation mechanism to the noisy, delayed rewards inherent in 2048. Our findings underscore the trade-off between deeper lookahead (increasing levels and simulation depth) and computational cost, with improvements observed up to level 3 at which even a shallow lookahead outperformed a level 2 configuration with a longer horizon. Overall, classic Monte Carlo and UCT produced adequate results, indicating that the randomness in tile placements partly diminishes the benefits of UCT's exploration-exploitation balance under a fixed-horizon setup.