

# Performance Measurement and Optimization Report

## 1. Introduction

This report aims to document the performance evaluation and optimization efforts applied to the Advanced TicTacToe Game developed in C++ with a graphical interface using Qt. The project includes AI logic based on the Minimax algorithm, user authentication, and session history management. Performance testing is crucial to ensure the game remains responsive, efficient, and stable, especially when handling AI computations.

## 2. Response Time Analysis

This section will cover the responsiveness of the game interface and AI processing.

- Player Move Response:

384, 718, 790, 1119, 1307, 809 → Average time = 854  $\mu$ s

- AI Move Time (Easy Mode):

489, 1071, 1910, 882, 1247, 1108 → Average time = 1117  $\mu$ s

- AI Move Time (Medium Mode):

11614, 1646, 468, 2416, 1577, 1410 → Average time = 3188  $\mu$ s

- AI Move Time (Hard Mode):

915, 1512, 1504, 2141, 960, 1593 → Average time = 1437  $\mu$ s

Measurement Method:

QElapsedTimer from Qt was used to capture the elapsed time of events such as handling player input and computing AI decisions.

## 3. Memory Usage

The game's memory usage is evaluated to ensure it stays within acceptable bounds during normal and AI-heavy scenarios.

- Memory Usage During player Move:

96.64, 96.754, 96.7734 → Average Memory Usage = 96.72 MB

- Memory Usage During AI Move:

96.652, 96.754, 96.785 → Average Memory Usage = 96.73 MB

Measurement Tool:

On Windows systems, GetProcessMemoryInfo was used to retrieve memory usage statistics for the application.

## 4. CPU Utilization

CPU usage helps determine the computational load, especially during AI processing.

- Idle Gameplay (Player vs Player):

Average = 0 % , Peak = 0.2 %

- Real Gameplay (Player vs Player):

Average = 0.3 % , Peak = 0.5 %

- Idle AI Mode (Hard Mode):  
Average =0.07 % , Peak =0.3 %

- Real AI Mode (Hard Mode):  
Average =0.3 % , Peak =0.8 %

"The reported CPU utilization percentages are approximate and rounded values, based on visual observation during runtime."

Observation Tool:

CPU metrics were monitored using Windows Task Manager and Qt Creator's built-in profiler.

## 5. Optimization Techniques

To improve performance and ensure smooth gameplay, the following optimizations are applied based on the current TicTacToe implementation:

- **Alpha-Beta Pruning in Minimax:**  
The minimax function is implemented to evaluate moves recursively. Although the current code uses a standard minimax approach, adding Alpha-Beta pruning would reduce the number of evaluated nodes by cutting off branches that cannot influence the outcome, improving efficiency.
- **Difficulty Level Controls:**  
The AI difficulty is controlled via predefined levels (`easyMove()`, `mediumMove()`, `hardMove()`) that adjust the complexity of the move search and the recursion depth indirectly by limiting the search or choosing moves heuristically, which helps balance performance and AI strength.
- **Move Calculation within UI Thread (No Multithreading Yet):**  
Currently, AI moves (`makeAIMove()`) are executed in the main thread. This is acceptable for a simple game like Tic Tac Toe but could cause minor UI blocking if the search tree is large. Multithreading (e.g., using `QThread`) can be considered to offload AI computation in future improvements.
- **State Reuse and Efficient Board Representation:**  
The board state is represented as a simple `std::vector<char>`, allowing easy and efficient copying for recursive evaluation in minimax. The code reuses this vector without unnecessary allocations, which improves memory efficiency.
- **Avoiding Redundant Calculations:**  
While no explicit memoization or caching is implemented in the current code, the structure of the minimax recursion with pruning (if added) would help reduce redundant calculations. Memoization could be added to further optimize performance.

## 6. Conclusion

Performance testing and optimization are essential in game development to deliver a responsive and efficient experience. Once real-world testing is completed and the measurements are filled in, this report will provide a clear view of the system's strengths and any remaining areas for improvement.