Performance Report

Project Overview

This report details the performance of the Advanced Tic Tac Toe Game, an embedded systems project developed using C++ and Qt. The game incorporates user authentication, personalized game history, and an intelligent AI opponent. Performance optimization and monitoring are key objectives, focusing on metrics such as response time, memory usage, and CPU utilization.

Performance Benchmarking Methodology

The performance benchmarking was conducted as part of the CI/CD pipeline using GitHub Actions. The `time -v` command was used to measure various system resources during the execution of the `./Embedded --version` command. This command provides detailed statistics on CPU time, elapsed time, memory usage, and page faults.

Performance Metrics Analysis

CPU Utilization

- **Percent of CPU this job got:** 6%

The CPU utilization metrics indicate that the `Embedded --version` command consumed negligible user and system CPU time, with a peak CPU usage of 6%. This suggests that the application\'s version check operation is not CPU-intensive.

Response Time

- **Elapsed (wall clock) time (h:mm:ss or m:ss):** 0:00.24

The elapsed wall clock time for the `./Embedded --version` command was 0.24 seconds. This indicates a very fast response time for this specific operation, which is crucial for a responsive user experience in an embedded system.

Memory Usage

- **Maximum resident set size (kbytes):** 19712

The maximum resident set size was 19712 kbytes (approximately 19.25 MB). This metric represents the maximum amount of physical memory (RAM) used by the process during its execution. For an embedded system, this memory footprint is relatively low, suggesting efficient memory management for the version check operation.

Page Faults

- **Minor (reclaiming a frame) page faults:** 961

Minor page faults occur when a process accesses a page that is in memory but not currently mapped to its address space. The 961 minor page faults indicate that the system had to perform some remapping of memory pages, but these do not involve I/O operations to disk. This number is not excessively high for the operation performed.

Conclusion

Based on the performance benchmarking of the `./Embedded --version` command, the application demonstrates efficient resource utilization with minimal CPU and memory overhead, and a very fast response time. The observed metrics are favorable for an embedded systems environment, indicating that the application\'s core functionalities are likely to perform well within resource constraints. Further performance analysis on more complex operations, such as game logic execution and AI decision-making, would provide a more comprehensive understanding of the overall system performance.