The artifact selected for enhancement is the GameService.java file, which implements a singleton service to manage game instances in a gaming application. Originally developed for a course project, this file demonstrates core software engineering principles such as encapsulation, abstraction, and the singleton design pattern. It manages unique IDs for players, teams, and games while ensuring that no duplicate game instances are created. I chose this artifact because it reflects foundational software design techniques, and I saw opportunities to improve its clarity, efficiency, and adherence to Java best practices.

In the enhanced version of this artifact, I made several key improvements. First, I simplified the addGame method by eliminating redundant logic and reusing the getGame(name) method to avoid code duplication. This made the code cleaner and more efficient. I also converted several static fields into instance fields, aligning better with proper singleton structure and encapsulation practices. Enhanced for-loops replaced index-based iteration for better readability and performance. I made sure to use final for the singleton instance and refined the in-line documentation to better describe each method’s purpose and logic.

Through this enhancement process, I reinforced my understanding of object-oriented principles, particularly the benefits of refactoring for readability and maintainability. I learned how important it is to structure code not just for functionality, but for clarity and scalability in larger systems. By applying better encapsulation and streamlining logic flow, I improved the reliability and structure of the code. This artifact enhancement supports several course outcomes, especially in software engineering design and demonstrating well-founded programming practices that align with industry expectations.