Michael Crevier
CS 499 Capstone
Milestone Four: Database Enhancement

Artifact Description

The artifact is a Python-based AI word strategy game that has evolved through three development phases. Enhancement 1 established a modular game engine, Enhancement 2 added AI prediction systems, and this final enhancement implements a robust database infrastructure for persistent learning and analytics. The enhancement introduces a SQLite database system with specialized repositories for word tracking, category management, and AI model data storage. The system features optimized data structures, transaction management, and a centralized DatabaseManager to coordinate operations while maintaining clean separation of concerns.

Justification for Inclusion

This enhancement demonstrates my ability to design and implement complex database architectures for real-world applications. The system showcases comprehensive database design through a normalized schema that efficiently captures gameplay data, player behavior, and AI learning metrics. I implemented proper indexing and transaction management to ensure both performance and data integrity. The repository pattern isolates data logic within specialized components, making the system more modular and maintainable. This architecture enables cross-session state management, allowing the AI models to learn from historical data and improve over time. The persistent storage transforms the AI from session-bound predictors into adaptive learning systems, enhancing both competitiveness and realism. Performance optimization through caching and efficient queries ensures responsive gameplay even as data volume grows.

Outcome Alignment

This enhancement aligns with the Databases outcome of the Computer Science program. The implementation demonstrates efficient schema design, cross-session data persistence, and proper transaction management. The database layer builds on the architectural foundation from Enhancement 1 and amplifies the AI capabilities from Enhancement 2. It serves as the backbone for persistent learning and long-term improvement, completing the three-phase development strategy outlined in Module One.

Reflection on Enhancement Process

The integration of a persistent database layer introduced several meaningful challenges. The most significant was designing a schema that could accommodate the game's evolving needs without sacrificing performance. I addressed this through normalization, careful relationship mapping, and strategic indexing. Another major challenge was integrating database functionality with the existing codebase. Although the repository pattern helped abstract low-level operations, coordinating the database layer with the game logic required significant refactoring. The AI models, originally designed for in-memory operation, needed careful adaptation for persistent data access.

The most complex aspect was managing transactions across multiple AI models. I encountered edge cases where improperly scoped transactions could lead to inconsistencies. Solving this required rigorous testing and restructuring to ensure atomic, isolated updates across all relevant repositories. Through this enhancement, I developed a deeper understanding of real-world database design, particularly in systems where data must evolve alongside the application. The experience reinforced the importance of considering persistence, recovery, and performance from project inception.

While the system continues to have areas for future improvement, it now demonstrates a clear and functional integration of database principles into a sophisticated AI application. The lessons learned throughout this process—from schema design to real-world integration—have been both challenging and rewarding, providing valuable insights for future database-driven applications.