CS 499 Computer Science Capstone

Daniel J. Vidmar

Southern New Hampshire University

CS499: Enhancement Three Narrative

Dr. Troy Hawk

Artifact Description

The third artifact presented for my ePortfolio is the MongoDB-based persistence layer developed for my 2D survival game project. This system enables the storage and retrieval of procedurally generated world data, including biome and terrain types, as well as player-specific information such as tile-based location and health state. This artifact was created and enhanced during CS 499 to showcase my skills in database integration and systems design within an interactive game environment.

I selected this artifact because it demonstrates practical experience in designing and implementing scalable data structures using a NoSQL solution tailored to a procedurally generated world. Through this enhancement, I built a WorldManager that caches tile data in memory and interacts with a WorldDataRepository to perform reads and writes to MongoDB. The system supports initialization on startup, generates default documents when missing, and integrates tightly with the game's terrain and biome generators. Player state is similarly persisted, allowing recovery between sessions.

This artifact aligns directly with several course outcomes, especially the ability to apply innovative techniques and tools in computing practices (Outcome 4) and the design of computing solutions using sound algorithmic principles and trade-off management (Outcome 3). Additionally, by considering the implications of system integrity and tamper resistance, this work touches on Outcome 5—developing a security mindset—even though full security measures have not yet been implemented.

A key insight gained from this work was the complexity of syncing procedural generation with persistent storage, particularly when supporting large worlds. Structuring data for efficient access, avoiding unnecessary writes, and handling cache invalidation were all important challenges I navigated during development. Another critical realization was the importance of thinking ahead about data security, especially in a networked environment.

Currently, the system does not enforce strict security on database writes—any process with access can potentially modify player or world data. For this academic prototype, this was acceptable; however, I recognize that for a commercial game product, robust server-side validation and restricted write access would be essential to prevent cheating and maintain gameplay integrity. Implementing role-based access controls, validating update logic server-side, and logging anomalies would be part of that future work.

This enhancement has also laid the groundwork for future features, including extending persistence to additional LivingEntity types such as animals or monsters. Each entity will eventually retain its own health state and activity logs across sessions. I also plan to implement player-world interaction tracking, such as tile edits, resource harvesting, or structure placement, as part of a more immersive and durable world state.

In summary, this artifact reflects my ability to design, build, and integrate a full-featured database solution within a performance-sensitive application. It shows not only my technical capability in managing structured data but also my awareness of scalability and security implications critical to modern software development, particularly in the context of games or interactive simulations.