M.U.D.

Yousaf, Graden, Luis, Nate, and Samuel.

Overall Experience:

- **Implementing the design was generally smooth, with most features, including user authentication, session storage, and data import/export, completed efficiently. Challenges included refining the "browse map" feature to limit early access and addressing errors in the game map, but transitioning to a JavaFX GUI was relatively straightforward.**

Retrospective Action Plan:

- Increased use of Discord improved task clarity, and a single additional meeting helped the team stay on track from R1 to R2. Instead of using Trello, Sam took on the role of clarifying requirements and coordinating tasks, acting as a "living Trello board."

What Went Well:

- The implementation of battles, shrines, and merchants proceeded smoothly with minimal debugging and only minor adjustments for integration. Transitioning to a GUI using JavaFX was also straightforward, as the existing code adapted well without significant rewrites.

Rough Spots:

- **In R1, the game used a single-tile room, but in R2, the Room class was refactored to Tile, and a new Room class was created. This update was challenging, causing multiple issues and requiring extensive debugging, which occupied most of Graden’s programming time.**

Updated Design:

- In R2, key design changes improved scalability and functionality, including separating Room and Tile to support complex map structures and adding a Shrine class for better encapsulation of shrine interactions. Adapting the game to JavaFX with MVC principles and introducing a DataManager class for save/load functionality streamlined the GUI and data handling, enhancing maintainability and potential future updates.

Status of the Implementation:

- Our current implementation meets key project requirements, including user authentication, data persistence, and interactive gameplay with map navigation. Future improvements will focus on enhancing map exploration, optimizing performance in map rendering, and addressing potential edge cases in inventory management and save/load consistency through further testing.

Reflection & Lessons Learned

- **The R2 phase taught us valuable lessons in collaboration, code design, and project management, with improved communication, incremental refactoring, and centralized requirement tracking enhancing our workflow. Balancing programming and documentation highlighted the need for role rotation to increase team members' hands-on experience, leaving us well-prepared and excited to refine and expand the game in future iterations.**

Demo:

- https://youtu.be/hv6V_oFkvio

M.U.D.

Yousaf, Graden, Luis, Nate, and Samuel.