Aisha Muftau and Ella Lind

Team Composition Analysis

1. **Approach of a Team Composed Solely of Computer Engineering Students**
A team made up entirely of Computer Engineering students would approach the dungeon-crawler project with a strong technical foundation in systems programming, computer architecture, and low-level software design. Because CE students are trained to think in terms of efficiency, hardware constraints, and optimization, their approach would likely emphasize:

a. Graphics and User Interface Implementation
   i. CE students would focus on performance-optimized graphics, likely choosing libraries such as SDL2 or OpenGL.
   ii. They would prioritize responsive rendering, efficient memory usage for textures, and smooth frame updates.
   iii. The interface design would be functional and optimized, though potentially less stylized compared to designs crafted by UX/UI specialists.

b. Run-Time Dungeon Generation
   i. Students would implement procedural generation algorithms.
   ii. The emphasis would be on correctness, computational efficiency, and deterministic reproducibility for debugging purposes.
   iii. They would likely use optimized data structures, such as adjacency lists or compressed arrays, to handle dynamic room layouts.

c. Adaptive Enemy Behavior
   i. They would design adaptive enemies using: Finite State Machines (FSM) or decision trees
   ii. Their implementation would be computationally efficient, reflecting CE training in algorithmic optimization.

The Overall Focus of a CE-only Team includes Efficiency, Low-level performance, Hardware awareness (possibly preparing the game to run on embedded systems), Clean and optimized C/C++ code, and functional, technically robust UI and graphics.

2. **Expanded Capabilities with Network Engineers Added to the Team**

Introducing Network Engineers significantly expands the scope and sophistication of the project. Their expertise allows the game to move beyond a single-player local experience into a connected, distributed system.

a. Online Multiplayer and Cooperative Play

   Network engineers can implement:

   i. Real-time player synchronization
   ii. Client-server architecture
   iii. Matchmaking systems
   iv. Cooperative dungeon exploration

b. Remote Dungeon Generation and Cloud Integration:

   Network engineers enable the dungeon generation logic to evolve in several ways, making the game more scalable and connected:

   i. Server-driven, allowing persistent world features.
   ii. Shared across multiple players in multiplayer sessions.
   iii. Stored and reloaded across devices, ensuring continuous gameplay

c. Adaptive Enemies Using Player Data:

   With networking support, enemy behavior can adapt dynamically using:

   i. Cloud-stored player behavior analytics
   ii. Cross-session tracking of each player's style
   iii. Global difficulty models informed by data from all players

Using these data sources, enemies can adapt not only to an individual player's tendencies, but also to collective strategies observed across the entire network.

d. Security, Protocol Design, and Data Integrity

   Network engineers ensure:

   i. Secure transmission of game state
   ii. Protection against cheating or spoofing
   iii. Real-time Updates such as Live leaderboards and events
   iv. Efficient communication protocols
   v. Seamless handling of disconnects, reconnections, or lag

Overall, the collaboration of CE and Network Engineers results in a more immersive, scalable, and robust game experience that goes far beyond the local single-player model. The Focus of a CE + Network Team is everything the CE-only team can do in addition to: Networked gameplay, Persistent worlds, Cloud-based AI adaptation, Secure client/server models, Real-time distributed game state handling.