Dungeon Crawler Game: Design, and Enhancements

1. Introduction

The goal of this project was to expand the provided dungeon-crawler game by adding progression, improved interactivity, and meaningful player choices. The assignment required four major gameplay enhancements:

- Progressive monster difficulty
- Increasing treasure value by depth
- Implementation of a coin collection system
- A flee mechanic with a coin-based penalty

The following sections describe how these features were incorporated into the game, along with additional improvements that enrich the player experience.

2. Progressive Monster Strength

To satisfy the requirement for difficulty scaling, the **createRoom()** function was modified so that monsters encountered deeper in the dungeon become stronger. The monster's power is calculated using the formula:

$monster\_strength = 15 + difficultyFactor$
$difficultyFactor = (index * 100) / numRooms$

Since index increases as the player moves deeper into the dungeon, the resulting monster strength increases accordingly. This creates a natural difficulty curve. The final chamber is designated as a **BOSS room**, ensuring a climactic final encounter.

3. Increasing Treasure Value With Depth

Treasure also scales with dungeon depth to reward players for taking on greater risks. The treasure calculation is:

$treasure\_value = 30 + difficultyFactor * 2$

This guarantees that treasure found in later rooms is more valuable, matching the progression of monster strength. Additionally, treasure rooms may contain randomized items such as weapons, armor, or potions, further increasing value and strategic depth.

4. Coin Collection System

A coins attribute was added to the Player structure: *int coins;*
Players earn coins through several actions:

- Looting treasure rooms
- Defeating monsters
- Completing boss battles

- Trading with merchants or selling items
  This satisfies the assignment requirement to track currency and update it throughout gameplay. The coin economy also supports the flee mechanic and merchant system.

5. Flee Mechanic With Penalty

The game allows players to either fight or flee when encountering a monster. To meet the project specification: *Flee cost = room level × 10 coins*

The flee system imposes a cost proportional to the room's depth. If the player cannot afford the flee fee, they are forced to fight the monster.
The final implemented logic is: *fleeCost = (roomIndex + 1) * 10;*
This ensures the penalty increases as the player delves deeper into the dungeon.

6. The following are the additional Enhancements we added Beyond Requirements just to make the game more strategic, engaging, and dynamic:

- Item system (weapons, armor, potions)
- Merchant rooms for buying and trading equipment
- Healing springs and trap rooms
- Experience system with leveling up
- Boss battle with multi-phase combat
- Save/Load system for persistent progress
- Graphical text formatting for immersion

7. Conclusion

The enhanced dungeon crawler successfully implements all required features: progressive difficulty, increasing rewards, a functional coin system, and a flee mechanic consistent with the assignment specification. Additional features significantly enrich gameplay and demonstrate thoughtful engaging design. The movement, combat, item systems, and adaptive rooms create a cohesive and enjoyable game world.