**1. Core Game Loop Module  
• Requirements List:  
o Room Switching Logic: Loading and switching between dungeon rooms (door triggers, teleportation conditions).  
o Combat Loop: Basic framework for player attacks, enemy hits, and skill cooldowns.  
o In-game Random Events: Random generation and trigger logic for treasure chests, traps, and shops.  
o Time Control: Global game speed adjustment (e.g., bullet time effect).**

**2. Character System Module  
• Submodule Requirements:  
o Character Attribute Management: Calculation of values for health, armor, movement speed, and energy (mana).  
o Skill System: Cooldown and effect triggering for active skills (e.g., mage fireball) and passive skills (e.g., health regeneration).  
o Character Animation: Frame animation switching logic for movement, attacks, damage, and death.  
o Pet/Companion System: Follow logic, auto-attack, and attribute inheritance.**

**3. Weapon and Item System Module  
• Submodule Requirements:  
o Weapon Type Library: Differentiation logic for melee, ranged, and special weapons (e.g., laser guns, shotguns).  
o Attack Mode Implementation: Ballistic calculations for fire rates, charge attacks, spread shots, rebounds, etc.  
o Weapon Dropping and Crafting: Random drop rules, weapon crafting formula (e.g., combining 3 identical weapons for an upgrade).  
o Item Effect Stacking: Temporary/permanent attribute modifications from buff potions and debuff curses.**

**4. Enemy and AI Module  
• Submodule Requirements:  
o Enemy Type Library: Differentiated behavior templates for melee minions, ranged shooters, and bosses.  
o Behavior State Machine: Logic for switching between patrol, pursuit, attack, and retreat states.  
o Bullet Hell System: Complex bullet patterns for bosses (e.g., circular spread, tracking shots, cross lasers).  
o Aggro Mechanism: Priority calculation for enemy target selection between the player and pets.**

**5. Dungeon Generation Module  
• Submodule Requirements:  
o Random Room Generation: Algorithms for room shape, size, and obstacle layout (e.g., BSP partitioning).  
o Level Theme Differentiation: Textures and traps associated with different themes like forest, glacier, volcano, etc.  
o Path Connectivity: Ensuring rooms are connected via corridors and there are no dead ends.  
o Dynamic Difficulty Adjustment: Dynamically increasing the proportion of elite enemies based on player progress.**

**6. Multiplayer Module**

**• Submodule Requirements:**

**o Network Synchronization: Real-time synchronization of player positions, states, and bullet trajectories (with latency compensation).**

**o Room Management: Creating/joining rooms, player matchmaking, and disconnection-reconnection mechanisms.**

**o Sharing Mechanism: Experience point distribution, item allocation rules (e.g., handling conflicts over collecting gold).**

**o P2P or Server Architecture: Choosing synchronization strategy (frame synchronization or state synchronization).**

**7. UI and Interaction Module  
• Submodule Requirements:  
o Main Menu System: Start game, character selection, settings, multiplayer entry.  
o Combat HUD: Health bar, mini-map, skill icons, weapon status display.  
o In-game Interaction Interface: Pop-up logic for opening chests, purchasing from shops, and selecting talents.  
o Dynamic Hint System: Floating tips for new player guidance and achievement unlocks.**

**8. Audio and Visual Module  
• Submodule Requirements:  
o Sound Effect Pool Management: Spatialized playback of weapon sounds, hit feedback, and environmental sounds.  
o Pixel Animation System: Frame management for pixel animations of characters, enemies, and bullets.  
o Lighting and Shadow Effects: Dynamic lighting (e.g., flashlight), explosion effects, and bullet trails.  
o Screen Shake: Camera shake effects during hits and explosions.**

**9. Data and Save Module  
• Submodule Requirements:  
o External Progression System: Persistent storage for gem currency, character unlocks, and skin purchases.  
o In-game Temporary Save: Save the current room state (enemy positions, item statuses) when paused.  
o Cloud Save Compatibility: Sync data through local storage (localStorage) or server.  
o Anti-Cheat Mechanism: Save file encryption and verification (e.g., MD5 hash check).**

**10. Performance Optimization Module**

**• Submodule Requirements:**

**o Object Pooling: Reuse management for bullets, enemies, and effects (to avoid frequent garbage collection).**

**o Collision Detection Optimization: Quadtree spatial partitioning, broad-phase collision checks first.**

**o Batch Rendering: Merge draw calls for textures with the same material (e.g., many bullets).**

**o Dynamic Loading: Load resources on demand (e.g., loading textures when entering a new room).**

**11. Scalability Module  
• Submodule Requirements:  
o Mod Support: External JSON configuration for weapon/character data and custom map templates.  
o Multilingual Support: Dynamic loading and switching of text resources.  
o Achievement System: Event listeners and achievement unlock conditions (e.g., killing 1000 enemies).  
o Daily Challenges: Random seed generation for special rule-based levels (e.g., only melee weapons allowed).**