**Phase 1: Planning and Conceptualization**

1. **Game Concept**:
   * story.
   * Sketch protagonist and key characters.
   * Determine features.
2. **Plan Core Systems**:
   * Combat system: Turn combat?
   * Character progression: Stats, leveling, and abilities.
   * Level up: +5 point to attribute
   * Inventory system: Items, equipment, and resource management.

**Phase 2: Setting Up Development**

1. **Environment Setup**:
   * Install JavaFX and set up your IDE (e.g., IntelliJ, Eclipse).
   * Create a Git repository for version control.
2. **Create a Project Structure**:
   * Example structure:
   * src/
   * - main/
   * - Game.java // Entry point
   * - assets/ // Resources (images, audio, etc.)
   * - models/ // Classes for characters, items, etc.
   * - ui/ // UI components
   * - systems/ // Combat, inventory, etc.
3. **Choose Tools**:
   * **Graphics**: Use pixel art tools like Aseprite for sprites or Tiled for map design.
   * **Sound**: Use free or custom soundtracks and sound effects.

**Phase 3: Prototyping Core Systems**

1. **Basic Game Framework**:
   * Set up a game loop in JavaFX.
   * Create basic scenes (e.g., title screen, overworld, battle screen).
2. **Character Class**:
   * Protagonist class with attributes.
   * Methods for leveling up, equipping items, and updating stats.
3. **Battle System**:
   * Start with basic turn-based combat:
     + Calculate damage using stats.
     + Implement attack, defend, and use item actions.
     + Add a simple enemy class for battles.
   * Expand to include special abilities and status effects.
4. **User Interface**:
   * Design health/mana bars.
   * Create menus for selecting actions in combat.

**Phase 4: Core Gameplay Mechanics**

1. **Overworld Exploration**:
   * Create a 2D map using JavaFX Canvas or integrate a library for tile-based maps.
   * Implement player movement with keyboard input.
   * Add NPCs and simple interactions.
2. **Inventory System**:
   * Design a UI for managing items.
   * Allow the protagonist to pick up, use, or equip items.
3. **Save/Load System**:
   * Use JSON or a database to save game progress (character stats, inventory, current location).

**Phase 5: Expanding Features**

1. **Advanced Battle Features**:
   * Add critical hits, type effectiveness, and elemental affinities.
   * Implement buffs, debuffs, and status effects like poison or paralysis.
2. **Enemy AI**:
   * Program simple enemy behavior (e.g., random attacks, healing).
   * Expand to more strategic AI (e.g., targeting weaknesses).
3. **Character Progression**:
   * Add experience points (XP) and leveling up.
   * Unlock new abilities or skills at higher levels.
4. **Quests**:
   * Implement a simple quest system:
     + Fetch quests: Collect items.
     + Defeat quests: Beat specific enemies.

**Phase 6: Content Creation**

1. **Design Maps**:
   * Create multiple areas (towns, dungeons, battle zones).
   * Populate maps with NPCs, enemies, and interactable objects.
2. **Write the Story**:
   * Flesh out the main quest and optional side quests.
   * Add dialogue for NPCs and key story moments.
3. **Art and Audio**:
   * Create or source character sprites, tilesets, and effects.
   * Add background music and sound effects for battles and exploration.

**Phase 7: Testing and Polishing**

1. **Test Core Mechanics**:
   * Ensure combat, movement, and menus work as intended.
   * Check for bugs or crashes in various scenarios.
2. **Balance Gameplay**:
   * Adjust stats for enemies and the protagonist.
   * Fine-tune damage formulas and difficulty progression.
3. **Improve the User Experience**:
   * Polish the UI for better navigation.
   * Add animations and effects for attacks and abilities.