Final Project: Dungeon Crawler Game Development

Estimated hours: 12-16 hours

Project Objective: The objective of this final project is to apply the programming concepts learned throughout the course to develop a text-based dungeon crawler game. Students will demonstrate their understanding of variables, arrays, structures, functions, pointers, linked lists, and file I/O by creating a complete game with character creation, inventory management, combat, and game persistence.

Pre-requisites:

- Knowledge of C programming fundamentals
- Understanding of arrays, structures, pointers, and linked lists
- Basic knowledge of file I/O operations

Project Tasks

Task 1: Game Character System (2 points)

- Create a structure called `Player` with the following properties:
    - o name (char array)
    - o health (int)
    - o maxHealth (int)
    - o attack (int)
    - o defense (int)
    - o experience (int)
    - o level (int)
    - o gold (int)
- Implement functions for:
    - o Creating a new player character (prompting for name and assigning default stats)
    - o Displaying player stats
    - o Leveling up a character (increase stats when experience reaches threshold)
    - o Healing a character (restore health but not exceeding maxHealth)

Task 2: Enemy and Combat System (2 points)

- Create a structure called `Enemy` with the following properties:
    - o name (char array)
    - o health (int)
    - o attack (int)
    - o defense (int)
    - o experienceReward (int)
    - o goldReward (int)
- Create an array of at least 5 different enemy types with varying statistics

- Implement functions for:
    - o Generating a random enemy appropriate for the player's level
    - o Combat mechanics (turn-based attacks between player and enemy)
    - o Determining battle outcome (victory, defeat, escape)
    - o Awarding experience and gold to player upon victory

Task 3: Inventory System (2 points)

- Create a structure called `Item` with the following properties:
    - o name (char array)
    - o type (int or enum) - use constants: 0 for WEAPON, 1 for ARMOR, 2 for POTION
    - o value (int) - represents attack bonus, defense bonus, or healing amount
    - o cost (int)
    - o description (char array)
- Implement a dynamic inventory system using a linked list
- Create functions for:
    - o Adding items to inventory
    - o Removing items from inventory
    - o Using items (applying effects to player stats)
    - o Displaying all inventory items
- Create a shop system where players can buy and sell items using gold

Task 4: Dungeon Generation and Game Loop (2 points)

- Create a simple dungeon structure represented by connected rooms
- Each room should have:
    - o A description
    - o Possible encounters (enemy, treasure, shop, empty)
    - o Directions to move (north, east, south, west)
- Implement the main game loop including:
    - o Player movement between rooms
    - o Random encounter generation
    - o Menu system for actions (move, check stats, access inventory, save game, quit)
    - o Win/lose conditions (reach a certain level or defeat a boss enemy)

Task 5: File I/O for Game Persistence (2 points)

- Implement save game feature that writes the current game state to a file:
    - o Player information
    - o Current inventory
    - o Game progress
- Implement load game feature that reads game state from a file
- Add error handling for file operations
- Create a menu system that allows players to:
    - o Start a new game

    - Load a saved game
    - Exit the game

Bonus Features (Optional)

- Implement a quest system with objectives and rewards
- Add special abilities that unlock at certain player levels
- Create a map system that reveals explored areas
- Implement different difficulty levels

Project Requirements

- All code must have meaningful variable/function names
- Program must compile without errors or warnings
- Program must handle invalid user input gracefully
- Menu systems should be clear and user-friendly