

Assignment: RPG Adventure Builder

This assessment is worth 50% of the overall marks for course CS1527.

You must submit your code and README via Codio before 23.59 on the 1st of May.

The markers will look at your code, and will visually check its output as well as in inspection of the internals. If the code does not run you may get a much lower mark.

Remember that as this is an individual assessment, work submitted must be your own.

If you haven't already done so, you should familiarise yourself with the University guidance on plagiarism, available at <https://www.abdn.ac.uk/sls/online-resources/avoiding-plagiarism/>. Also note that, use of automated services or AI to generate submissions for assessment will be treated as academic misconduct and pursued under the University misconduct procedures.

If you reuse other people's code or libraries, you must explicitly mention this in your code and acknowledge their work. This includes: - Where did you acquire them (e.g. URLs, books, GitHub)? - Who are the authors? They cannot be your classmates. - How did you modify them? In particular, if you make minor changes to a piece of other people's code, you must clearly mark in your code which part is your own code.

Overview

In this project, you will receive a starter code package for a basic terminal-based role playing game (RPG) game. The provided code includes essential elements such as character creation, a simple combat routine, a basic inventory system, and a main game loop. Your task is to build upon this foundation by adding new features, improving gameplay mechanics, and polishing the overall user experience.

A glossary of RPG specific terms that those not familiar with games is included. However, if anything is not clear, please email me or speak to my at practical sessions.

Starter Code Package

The starter code includes: - **Game Loop & Main Menu:** A basic loop that displays a menu for actions like "Start Game" and "Exit". - **Character Class:** A simple class defining a player with attributes such as name, hit points, and an inventory list. - **Combat Module:** A rudimentary combat function that pits the player against a single enemy with fixed stats. - **Inventory System:** Functions to add, remove, and display items.

You may change the starter code in any way that you deem fit.

Core Requirements

1. **Enhance the Combat System - 10 points**
 - **Multiple Enemy Types:** Extend the combat module to include at least two different enemy classes with varying behaviors and stats.
 - **Expanded Combat Options:** Allow players to choose between several actions (e.g., attack, defend, use an item, special moves) during combat.
 - **Dynamic Encounters:** Introduce elements such as critical hits or dodges using random chance to make battles less predictable.
2. **Extended Item Statistics and Combat Loop Enhancements - 10 points**
 - **Item Attributes:** Modify items so that they now include:
 - **Attack Stats:** Increases the damage dealt when the item is equipped.
 - **Defence Stats:** Reduces the damage taken when the item is equipped.

- **Combat Integration:** Update the combat loop to consider these new item statistics when calculating both damage dealt and damage taken, creating a more strategic and dynamic battle experience.
3. **Introduce Drops - 10 points**
 - **Enemy Drops:** Implement a system where defeated enemies have a chance to drop items.
 - **Drop Chance and Rarity:** Assign drop probabilities based on enemy type and item rarity, ensuring that more powerful items are less common.
 - **Inventory Integration:** Add dropped items to the player's inventory, with clear notifications during gameplay.
 - **Loot Management:** Provide players with the ability to inspect, accept, or decline dropped items, enhancing the strategic element of loot management.
 - **Randomised Attributes:** Include randomised attributes for dropped items to add variability and replayability to encounters.
 4. **Implement Consumables - 10 points**
 - **Consumable Items:** Modify the existing inventory system to include consumable items (such as health potions, mana potions, or temporary stat boosters).
 - **Usage Mechanics:** Allow players to use these items during combat or exploration to regain health, restore mana, or temporarily boost stats.
 - **Usage Constraints:** Implement rules such as cooldowns, limited quantities, or usage restrictions to prevent overuse.
 5. **Expand the Inventory and Introduce Crafting - 10 points**
 - **Enhanced Inventory Display:** Improve the inventory system to display item details like names, descriptions, and categories.
 - **Crafting Mechanics:** Let players combine certain items to craft new equipment or consumables. For example, combining "Herbs" and a "Vial" might create a "Health Potion."
 - **Item Validation:** Ensure that item combinations follow logical rules and provide feedback if the crafting attempt fails.
 6. **Save/Load Functionality - 10 points**
 - **Persistent Game State:** Add the ability to save and load the game state, capturing character stats, inventory contents, current quest progress, and explored areas.
 - **Robust File Handling:** Ensure your implementation gracefully handles file errors (e.g., missing files, corrupted data).
 7. **Implement Unit Tests - 10 points**
 - **Comprehensive Testing:** Develop unit tests for key modules of the game (such as combat, consumables, crafting, and save/load functionality) using either unittest or pytest.
 8. **High Quality Coding Standards - 10 points**
 - **Code Comments and Documentation:** Ensure that all code is well-commented and includes clear documentation to explain the functionality of each module, class, and function.
 - **Modular Design:** Split functionality across multiple files (e.g., separate modules for combat, inventory, utilities, and the game loop) to promote readability and maintainability.
 - **Coding Best Practices:** Follow high-quality coding standards, including proper naming conventions and error handling.
 9. **Task 9: README Documentation - 20 points**
 - Create a comprehensive README file for your RPG adventure game. This file must be submitted on Codio and clearly document the following aspects of your project.
 - **How to Run the Game:**
Provide clear, step-by-step instructions for launching and playing your game.
 - **How to Run the Unit Tests:**
Include instructions on how to execute the unit tests (using unittest or

pytest).

- **Task Approach Summary:**
Summarise your approach to each task, outlining your design decisions and methodology.
- **New Features:**
List and describe all the new features you have implemented throughout the project.
- **Challenges and Solutions:**
Explain any challenges you encountered during development and how you solved them.

Deliverables

- **Source Code:** Submit the complete, well-documented source code on codio. You may develop elsewhere but ensure that everything is working on codio for submission. Please do not include unrelated files.
- **README File:** Include a README that explains (readme should be submitted on codio):
 - How to run the game.
 - How to run the unit tests.
 - A summary of how you have approached each task.
 - The new features you've implemented.
 - Any challenges you faced and how you solved them.
 - README can be in .md format or .pdf

Glossary of RPG Specific Terms

- **Main Game Loop:** The continuous cycle that drives the game—updating game states, processing inputs, and displaying outputs to keep the game running.
- **Consumable Items:** Items that can be used once to grant benefits (such as restoring health or providing temporary boosts) and then disappear from the inventory.
- **Crafting:** The process of combining multiple items or materials to create a new, often more powerful or useful, item.
- **Enemy Drops:** Items that enemies may leave behind when defeated, which the player can collect as rewards or loot.
- **Item Rarity:** A system that categorises items based on how common or rare they are; rarer items are typically more powerful or valuable.
- **Mana:** A resource used to perform magical abilities or special moves. It is often represented numerically, depletes with each use, and can be restored with potions or resting.
- **RNG (Random Number Generator):** A tool used to introduce randomness into the game, affecting aspects like critical hits, dodge chances, or the likelihood of enemy drops.
- **Turn-Based Combat:** A combat system where players and enemies take turns performing actions, rather than all actions happening simultaneously.

Marking Criteria

1. Combat System Enhancements (10 points)

- **Multiple Enemy Types (4 points)**
 - **4 pts** - Two or more enemy types with unique stats and behaviors.
 - **3 pts** - Two enemy types, but variations are minimal.
 - **2 pts** - Two enemies exist but lack unique behaviors.
 - **1 pt** - Only one enemy type is present.
 - **0 pts** - No additional enemy types.
- **Expanded Combat Options (3 points)**

- **3 pts** - Multiple options (attack, defend, item, special move) that affect gameplay.
- **2 pts** - Options exist but are unbalanced or not fully functional.
- **1 pt** - One additional option beyond basic attack.
- **0 pts** - No additional options.
- **Dynamic Encounters (3 points)**
 - **3 pts** - Critical hits, dodges, or other randomness significantly impact combat.
 - **2 pts** - Some random elements exist but are predictable or limited.
 - **1 pt** - A random factor exists but does not affect gameplay meaningfully.
 - **0 pts** - No dynamic encounters.

2. Item Statistics & Combat Loop Enhancements (10 points)

- **Item Attributes (5 points)**
 - **5 pts** - Items have well-integrated attack/defense stats affecting combat.
 - **4 pts** - Stats exist but are not well-balanced.
 - **3 pts** - Some items have stats, but they are not used correctly in combat.
 - **1-2 pts** - Basic implementation with minimal impact.
 - **0 pts** - No additional item stats.
- **Combat Integration (5 points)**
 - **5 pts** - Item stats affect damage calculations and strategy.
 - **4 pts** - Stats influence combat but not consistently.
 - **3 pts** - Basic effects but not properly balanced.
 - **1-2 pts** - Implemented but ineffective.
 - **0 pts** - No item integration in combat.

3. Enemy Drops System (10 points)

- **Enemy Drops (4 points)**
 - **4 pts** - Enemies drop items consistently.
 - **3 pts** - Drops exist but are inconsistent.
 - **2 pts** - Drops work for only certain enemies.
 - **1 pt** - Items sometimes drop, but not correctly.
 - **0 pts** - No drop system.
- **Drop Chance & Rarity (3 points)**
 - **3 pts** - Drop rates vary based on item rarity and enemy type.
 - **2 pts** - Drop chances are defined but unbalanced.
 - **1 pt** - Basic drops, no consideration of rarity.
 - **0 pts** - No drop chance system.
- **Loot Management (3 points)**
 - **3 pts** - Players can inspect, accept, or decline items.
 - **2 pts** - Items drop but lack management options.
 - **1 pt** - Items are added directly without player choice.
 - **0 pts** - No loot management.

4. Consumables System (10 points)

- **Consumable Items (5 points)**
 - **5 pts** - Multiple consumables with clear purposes.
 - **4 pts** - At least two working consumables.
 - **3 pts** - One type of consumable implemented.
 - **1-2 pts** - Consumables exist but are ineffective.
 - **0 pts** - No consumables.
- **Usage Mechanics (5 points)**
 - **5 pts** - Players can use consumables in combat/exploration with proper constraints.

- **4 pts** - Works but may have balancing issues.
- **3 pts** - Usable but missing mechanics (e.g., cooldowns).
- **1-2 pts** - Implemented but not well-integrated.
- **0 pts** - No usable consumables.

5. Inventory & Crafting System (10 points)

- **Enhanced Inventory Display (4 points)**
 - **4 pts** - Shows names, descriptions, and categories clearly.
 - **3 pts** - Display works but lacks organisation.
 - **2 pts** - Basic inventory with minimal details.
 - **1 pt** - Inventory exists but is difficult to use.
 - **0 pts** - No inventory display.
- **Crafting System (4 points)**
 - **4 pts** - Functional crafting system with clear item combinations.
 - **3 pts** - Crafting works but lacks validation.
 - **2 pts** - Limited crafting, few possible recipes.
 - **1 pt** - Basic concept but not functional.
 - **0 pts** - No crafting.
- **Validation & Feedback (2 points)**
 - **2 pts** - Players receive feedback on failed crafting attempts.
 - **1 pt** - Limited feedback.
 - **0 pts** - No validation.

6. Save/Load Functionality (10 points)

- **Persistent Game State (5 points)**
 - **5 pts** - Saves/reloads all necessary game data.
 - **4 pts** - Saves most data, but minor issues exist.
 - **3 pts** - Some progress is saved, but major gaps exist.
 - **1-2 pts** - Save exists but is unreliable.
 - **0 pts** - No save/load function.
- **Robust File Handling (5 points)**
 - **5 pts** - Gracefully handles missing/corrupt files.
 - **4 pts** - Some error handling present.
 - **3 pts** - Basic error handling.
 - **1-2 pts** - Barely any handling.
 - **0 pts** - No file handling.

7. Unit Testing (10 points)

- **Comprehensive Testing (10 points)**
 - **10 pts** - Extensive tests for all major systems.
 - **7-9 pts** - Good test coverage, missing edge cases.
 - **5-6 pts** - Some tests but lacks coverage.
 - **2-4 pts** - Few tests, incomplete.
 - **0-1 pts** - No tests.

8. Code Quality & Documentation (10 points)

- **Code Comments & Documentation (5 points)**
 - **5 pts** - Clear comments throughout the code.
 - **4 pts** - Mostly well-documented.
 - **3 pts** - Basic comments exist.
 - **1-2 pts** - Minimal comments.
 - **0 pts** - No comments.
- **Modular Design & Best Practices (5 points)**

- **5 pts** - Well-structured, modular code.
- **4 pts** - Good organisation, some inconsistencies.
- **3 pts** - Some modularity, needs improvement.
- **1-2 pts** - Poorly structured, difficult to read.
- **0 pts** - Poorly organised.

9. README (20 points)

- **Content (10 points)**
 - **10 pts** - Explains setup, game features, and challenges clearly.
 - **7-9 pts** - Covers most details but lacks depth.
 - **5-6 pts** - Includes basic instructions.
 - **2-4 pts** - Minimal explanations.
 - **0-1 pts** - No README or unclear.
- **Clarity & Organisation (5 points)**
 - **5 pts** - Well-structured and easy to read.
 - **3-4 pts** - Mostly clear.
 - **1-2 pts** - Difficult to follow.
 - **0 pts** - Confusing or missing.
- **Formatting & Professionalism (5 points)**
 - **5 pts** - Free of typos, well-formatted.
 - **3-4 pts** - Minor errors.
 - **1-2 pts** - Poor formatting.
 - **0 pts** - Unprofessional.