



Assignment 3: (CLO-2 & CLO-3)

[Capstone Project: Text-Based RPG - "Quests of Eldoria"]

Problem

Welcome to the mythical land of Eldoria, where brave adventurers embark on quests to become legendary heroes. In this console-based RPG, players create a character, explore the world, engage in battles, and complete quests. The game features dynamic storytelling, character progression, and a variety of quests to keep players immersed in the fantasy realm.

Features:

1. *Character Creation:*
 - a. Allow players to create a character by entering a name and choosing a class (Warrior, Mage, Rogue).
 - b. Initialize the character's health and experience points.
2. *Display Character Information:*
 - a. Create a function to display the player's character information, including name, class, health, and experience points.
 - b. Show this information at the beginning of each quest.
3. *Embark on Quest:*
 - a. Design a quest system with different difficulty levels.
 - b. Present players with quest options, and let them choose which quest to embark on.
4. *Battle System:*
 - a. Implement a turn-based battle system against mythical creatures.
 - b. Use random number generation to simulate attacks and determine outcomes.
 - c. Manage player health and enemy difficulty levels.
5. *Leveling Up:*
 - a. Implement a leveling system based on experience points.
 - b. When the player accumulates a certain amount of experience points, allow them to level up.
 - c. Increase the player's health and reset experience points after leveling up.
6. *Dynamic Storyline:*
 - a. Create a dynamic storyline with branching paths based on player decisions during quests.
 - b. Use conditional statements to adapt the narrative based on player choices.
7. *User Input for Quest Choice:*
 - a. Allow players to choose quests by entering corresponding numbers.
 - b. Use switch statements to handle different quest choices.
8. *Game Loop:*
 - a. Implement a game loop that continuously allows players to explore and engage in quests.
 - b. Allow players to exit the game at any time.
9. *Inventory System (Extra Credit):*

- a. Implement a basic inventory system using arrays.
- b. Allow players to collect items during quests and display their inventory.

Implementation Guidelines:

1. Use structures for representing the player and quests.
2. Functions should be employed for various functionalities, ensuring a modular and organized code structure.
3. Utilize arrays for implementing features like inventory (optional) or storing quest information.

Additional Challenges (Extra Credit):

1. Implement a scoring system based on quest completion and performance in battles.

Getting Started:

1. Begin by designing structures for the player and quests.
2. Implement functions for character creation, quest embarkation, battling, leveling up, and displaying character information.
3. Gradually introduce features such as the quest system, dynamic storyline, and random events.
4. Use arrays for managing inventory or storing quest information, depending on complexity.

Deliverables:

1. **Design:**
 - a. Flowchart for the Application
2. **Development:**
 - a. Working C++ application with Code.

Instructions:

- This is a group assignment (maximum 4 members in a group).
- **You will need to present the demo for the application and be prepared for a viva. If you cannot explain your code and answer questions from the evaluators, you will not receive marks.**
- **All code and artifacts are to be uploaded on a public Github repository.**
- **Deadline: 4th January, 2024 (Vivas will be conducted)**
- Come up with a name for you application. (Bonus mark)