

Final Project: Text based Game

Outline

In this assignment you will build a text based RPG game using C++.

This project piggybacks from Assignment 2.

Use the short story you came up with in assignment 2. In your game; the story must include an **introduction** that describes the world the game takes place, as well as **action scenarios** the player will face. The story should have a **clear quest for the player** and add at least **3 options for player customization** (name, class, sex for example).

In your game include a player inventory system and all the components within it. That includes **player item list, display list, equip, unequip, use item, and discard item(s)** must be included.

Include a functioning merchant system. That includes methods or functions that will: **display merchant item list, check item availability, allow merchant exchange (buying/selling), allow merchant to set price, and allow player to exchange goods (buying/selling).**

In your game include at least **3 options** the player can make that impacts the story. That may be anything that changes the course for the player.

Include a combat system. That includes functions for **player display health and damage, player win/loss condition** for combat, **enemy display health and damage**, at least **3 enemy classes using inheritance**, and of course a **clear combat system** with randomization.

Also include a win/loss condition, that includes functions for **clear win objectives** and **loss conditions** for player, and **clear conditions using if/else states** and/or **switch statements**.

READ THIS it's worth 10 marks.

The very first part of this assignment is to setup a "RPG Game / C++ Project" **private** git repository on GitHub and add your Instructor as a collaborator.

Grading Rubric:

Requirements:

- GitHub repository setup with instructor added as a collaborator - (ALL THE MARKS)
- A short story for your game (25 marks)
 - including:
 - An intro describing the world - 5 marks
 - Action scenarios the player will face - 10 marks
 - A quest or purpose for the player - 5 marks
 - Functions for customizing player
 - include at least 3 customizations (name/class/sex for example) - 5 marks
- Functions in place for player inventory system (20 marks)
 - including:
 - Player item list - 5 marks
 - Player display list - 5 marks
 - Player equip/unequip/use/discard items - 10 marks
- Functions in place for merchant system (40 marks)
 - including:
 - Merchant item list - 5 marks
 - Merchant display list - 5 marks
 - Merchant item availability - 5 marks
 - Merchant exchange (buying/selling) - 10 marks
 - Merchant set price - 5 marks
 - Player exchange (buying/selling) - 10 marks
- Functions in place for game choices (30 marks)
 - including:
 - At least 3 options the player can make that impacts the story - 30 marks
- Functions in place for a combat system (50 marks)
 - including:
 - Player display health and damage - 5 marks
 - Player win/loss condition for combat - 5 marks
 - Enemy display health and damage - 5 marks
 - At least 3 enemy classes using inheritance - 15 marks
 - Clear combat system including randomization - 15 marks

- Functions in place for a win/loss condition (20 marks)
 - including:
 - Clear objectives for player - 5 marks
 - Clear conditions using if/else states and/or switch statements - 15 marks

Above Requirements are out of 155 marks

Coding Standards:

- Code indentation / formatted properly - 5 marks
- Good clear naming of variables - 10 marks
- Good use of Functions - 5 marks
- Good use of conditional blocks and case statements (if/else-if/else/switch) - 5 marks
- Good use of loops (do/do-while/while/for) - 5 marks
- Code compiles without errors or warnings - 5 marks
- Use of pointers in place of passing by copy (will make sense next week) - 10 marks

Above Standards are out of 45 marks

Assignment is out of 200 marks total. (we expect you to hit at least a 140, but please aim for 200)

Optionally (read best practice), you should setup a Kanban board (Trello) for yourself or team. Take the above features / requirements and break them down into small tasks that you can focus on. Planning what feature you will implement next as you go. Moving the cards across the board is rewarding and should help to keep you focused and motivated.

GLHF