

Dungeon

Are you ready for an adventure?

Outline

1. Introduction
2. Requirements
 - Basic Functions
 - Optional Enhancement
 - Report
 - Video
3. Grading
4. Submission
5. Get Help

Introduction

In this homework, you will make a text-based RPG with the player exploring a dungeon and fight against unknown monsters.

You will implement it using C++ and put your OOP knowledge into practice. Header files and tutorial videos are provided. Try your best to accomplish it.

Let's go for an adventure! >>

Requirements

1. Basic Functions (70%)
 - Movement (10%)
 - Showing Status (5%)
 - Pick up Items (10%)
 - Fighting System (10%)
 - NPC (10%)
 - Game Logic (10%)
 - Character Class Design(10%)
 - Trading System (5%)
2. Optional Enhancement (10%)
3. Report (15%)
4. Video (15%)

Actions Menu

Allowing player to choose next action.

What do you want to do ?

- A. Move
- B. Check Status
- C. Talk to Shop
- D. Open Backpack

Movement(10%)

Allowing player to move from one room to the other.
e.g. Moving from Room No.1 to Room No.2, and then
go back to Room No.1

Where do you want to go?

- A. Go up
- B. Go down
- C. Go left
- D. Go right

Showing Status(5%)

Dump out the player' s information.

Status:

[Player's Name]

> Health: 100/100

> Attack: 30

> Defense: 0

> Items: (if you have)

> (All you want to add...)

Pick up Items(10%)

Allowing user to pick up items and equip them onto the player.

Fighting System(10%)

Player can attack the monster, and the monster will either die or fight back if it is still alive. Moreover, the fight will loop until either one of them dies or the player retreats.

- Attack (kill the monster or the monster will fight back)
- Retreat (send player back to the previous room)

NPC(10%)

- Showing the script of NPC (what to say from the NPC)
- Trade with the player (player can get item(s) from the NPC)

Game Logic(10%)

Handle when the player win or lose.

Character Class Design(10%)

Adding Character Class Design.

Note that the UML does not include the design for a “Character Class Design”. You need to make up a design for coding by yourself.

The Character Class Design should include three types of items.

- Occupation
- Skills, Ultimate Abilities
- Corresponding Weapons

Add more item types as you wish, and show your advanced design in Optional Enhancement of the report.

More information on the next page.

Character Class Design

Occupation

- Must design at least 3 occupations. (ex: Warrior, Rogue, Magician...)
- Each occupation requires different physical characteristics, such as blood, attack and defense capabilities, and each occupation will have its own special outstanding parameters.

Skills, Ultimate Abilities

- Different occupations will have different skills.
- Design Buff and Debuff skills to apply different effects to your own character, an ally's character, or an enemy's character.

Corresponding Weapons

- Weapons need to be matched to the correct occupation to provide full effect.
- Design how to get weapons, e.g., from fighting a monster, opening a reward chest, or talking with NPC.

Buff and Debuff means: <https://smallgiantgames.helpshift.com/hc/en/4-empires-puzzles/faq/331-what-are-buffs-debuffs-status-effects/>

Trading System (5%)

Trading Process

- Design the way of trading with the NPC in the game.
- Determining the value of various items.
- Must design at least 1 additional way to spend the money. (ex. Tolls)

Money

- Design how to get money, e.g., from fighting a monster, opening a reward chest, or talking with NPC.

Optional Enhancement(10%)

You are allowed to add additional functions, features or even visualize the gameplay to enhanced the game experience (as long as the system works).

- Please describe the additional features clearly in your demo video and especially in your paper report.
- Example
 - MP system
 - Cooling Down Time (CD)
 - Type counter (i.e. water counters fire, fire counters grass, grass counters water)
 - DPs, tank, etc.
 - Record System(Save/Load the game)

Note

1. Our template is meant to speed up your development not to constraint your creativity. You can modify it or add extra functions whenever you want (as long as it still fulfill all the requirements).
e.g. You might change the return type of a function.

2. In both demo video and paper report, all types of rooms should exist in the map in order to demonstrate how you fit all the requirements.
e.g. Rooms of 1. monster, 2. NPC, 3. chest, 4. exit (boss).

3. You are allowed to design your own text-based game (complete different with our dungeon), but you must implement with OOP structure and describe in detail the OOP framework you are using in the report. (We will inspect your game more strictly than dungeon type.)

Grading

1. You can get the **basic score (70%)** after finishing all the basic requirements.
2. You might get a **higher score (up to 110%)** if you successfully enhance your dungeon game with other functions or abilities.
3. You must implement the assignment with **inheritance** and **virtual functions**, or there will be a **0 score**. **Please describe the structure of your inheritance and virtual functions clearly in the paper report.**

Grading

4. There will be **no demo section** but **a demo video** and **a detailed paper report** instead.
5. Optionally enhanced functions or features should also be described clearly so we can easily get your ideas and know your efforts.
6. About video (15%):
 - The video should be **no more than 300 seconds**.
 - The video should demonstrate all your working functions and **describe your UML design approach**.
 - **Editing is needed, add some labels to emphasize your features.**
(But faking your results are not allowed. We will inspect your video strictly to follow your code.)

Grading

7. About report (15%):

The report should be **clear and organized** .

- The report should describe **how you fulfill the requirements** .
- The report must contain the following:
 - Implementation detailed
 - UML Design
 - Results
 - Discussion
 - Conclusion

8. **No plagiarism! Any plagiarism will get 0 score.**

Submission

1. A zip file named <HW1_studentID.zip>
 - A folder of all your codes and resources named <Dungeon_studentID>
 - A paper report named <report_studentID.pdf>
 - A demo video named <demo_studentID.mp4>
 - You will get 0 score if any file missed.
3. Any wrong submission format will be 5% penalty.
3. Uploaded onto E3 platform before 4/14 (Fri) 11:59 pm
4. Late penalty will be 10% per day .

Get Help

Tutorial

The playlist may help you.

<https://www.youtube.com/playlist?list=PLoISxpCB-u1AkcGChEgFomFRhuLrFsOCu>

Any question

- Lab time: Tuesday 6:30(p.m.) ~ 9:30(p.m.) EC315
- New e3 Forum: Others may have the same question.
- E-mail: cc to all TAs