Dungeon

Are you ready for an adventure?

Outline

- Introduction
- Requirements
  - Basic Functions
  - Optional Enhancement
  - Report
- Grading
- Submission
- 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

- Basic Functions (80%)
  - Movement (15%)
  - Showing Status (5%)
  - Pick up Items (10%)
  - Fighting System (15%)
  - NPC (10%)
  - Game Logic (10%)
  - Backpack System (15%)
- Optional Enhancement (10%)
- Report (10%)

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(15%)

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)

Pick up Items(10%)

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

Fighting System(15%)

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.

Backpack System (15%)

Adding a Backpack System.

Note that the UML does not include the design for a Backpack System.
You need to make up a design for coding by yourself.

The Backpack System should include three types of items.

- Equipment
- Potion
- Key

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

More information on the next page.

Backpack System

Equipment

- Equipment can be put on to head, body, left/right hands.
- Each body part can equip 1 equipment.
- A replaced equipment on the same body part will be put back in inventory.

Potion

- Using potion on a player heals the player's health.
- Using potion on a monster damages the monster's health.
- Potion vanishes after being used.

Key

- Rooms that are locked can be opened with a key.
- Design how to get a key, 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
  - Skills
  - Ultimate Abilities
  - MP system
  - 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 youfit all the requirements.
   e.g. Rooms of 1. monster, 2. NPC, 3. chest, 4. exit (boss).

Grading

1. You can get the basic score (80%) after finishing all the basic requirements.
2. You might get a higher score (up to 100%) 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 20% penalty. 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:

- The video should be no more than 180 seconds .
- The video should be clear and demonstrate all you working functions.
- Editing is allowed (but faking your results are not allowed).

Grading

7. About report:

The report should be clear and organized .

- The report should describe how you fulfill the requirements .
- The report should contain the following:
  - Implementation detailed
  - 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.
2. Any wrong submission format will be 5% penalty.
3. Uploaded onto new E3 platform before 4/22 (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=PLolSxpCB-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