

CS 246 Assignment 5 - CC3k – mpyoung

How can you design your system so that each race can be easily generated? Additionally, how difficult does such a solution make adding additional classes?

To make each class be easily generated, we can use the Decorator pattern for Players and a Decorator pattern for each NPC. To add additional classes, all we need to do is create more subclasses extending the Player class.

How does your system handle generating different enemies? Is it different from how you generate the player character? Why or why not?

To generate different enemies, we can use the Factory pattern. It will be different from the player generation because there is only one player character. So for the player, we will use the singleton pattern.

How could you implement the various abilities for the enemy characters? Do you use the same techniques as for the player character races? Explain.

To implement various abilities, we use the Decorator pattern to make different attack functions for each class. We can do the do exact same thing with the player character races and make different attack functions for each player class.

What design pattern could you use to model the effects of temporary potions (Wound/Boost Atk/Def) so that you do not need to explicitly track which potions the player character has consumed on any particular floor?

Since there are only 3 types of potions, we can store the cumulative effects. We only need to store the cumulative attack and defense boost. For HP potions, we can just simply add or subtract the current hp.

How could you generate items so that the generation of Treasure and Potions reuses as much code as possible? That is, how would you structure your system so that the generation of a potion and the generation of treasure does not duplicate code?

We can generate potions and treasures at the same type but the only difference will be the type that is created. We will use the same methods for generating random coordinates on the game.

Project Breakdown Expectations:

Setting up base framework of game – 1 day

- Loading map from file
- Simple interpreter
- Creating class template files for player and NPCs
- Basic game class

Setting up player class – ½ day

- Make it easy to extends to Player class
- Implement double dispatch for Player class

Setting up base for monsters – ½ day

- Make it easy to extend new NPC classes
- Implement double dispatch for NPCs

Adding potions and treasures – ½ day

- Add different potion types
- Add treasure types

Extending all enemy and player classes – 1 day

- Implement player classes
- Implement all enemy abilities

Adding last touches – 1 day

- Add win/lose conditions
- Add multiple levels

QA and bug fixing – ½ day

- Test for bugs and ensure everything works

Additional DLC if time permits