

Tema POO - World of Marcel

Nume si prenume: Ciocan-Merauta Mihai

Grupa : 324CC

Dificultate : medie

Timp alocat rezolvarii : Am lucrat constant pe toata perioada vacantei de iarna.

Clase si interfete folosite:

1. Game
2. Information
3. Credentials
4. Grid
5. Cell
6. CellElement
7. Entity => {Character, Enemy}
8. Enemy
9. Character => {Warrior, Rogue, Mage}
10. Warrior
11. Rogue
12. Mage
13. Spell => {Ice, Fire, Earth}
14. Inventory
15. Potion
16. HealthPotion
17. ManaPotion
18. Shop
19. Visitor
20. Element
21. firstPage
22. lastPage

Explicatii pentru clasele interesante/importante:

1. Game:

Clasa principala, instantiata o singura data folosind Singleton.

Metode folosite:

run:

Parsez conturile si povestile din fisierele JSON.

Deschid fereastra de login, iar dupa autentificare incep sa iau input de la utilizator (urmatoarele mutari)

Verific fiecare pozitie a caracterului, iar daca in pozitia pe care se afla caracterul exista un SHOP, ENEMY sau a ajuns la FINISH, iau diferite decizii.

In cazul SHOP-ului, aleg daca sa cumpar o potiune sau nu, iar daca o cumpar afisez si inventarul curent.

In cazul unui ENEMY, aleg daca sa ma lupt cu el sau nu, iar daca da, fiecare entitate are cate un turn in batalie. Pot alege sa ma lupt cu pumnul sau cu abilitati (in limita manei disponibile). Dupa fiecare pierdere de viata sau miscare, printez statusurile entitatilor (in

functie de cine ataca).

In cazul FINISH-ului, salvez progresul in fisiere JSON si printez un mesaj corespunzator.

JSONParse:

Parsez fiecare poveste si fiecare cont din fisierele JSON folosindu-ma de un foreach (pentru fiecare obiect din JSON) si de metodele parseStories si parseAccounts

parseStories:

Pentru fiecare tip de celula, iau cate o poveste si o bag in lista corespondenta fiecarei celula din hashmap-ul de celule.

parseAccount:

Parsez din fiecare obiect JSON informatiile contului din obiectul respectiv, creez cate un cont pentru fiecare si adaug in lista de conturi.

writeProgress:

Metoda aceasta foloseste principiul metodei parseAccount, numai ca invers. Pentru fiecare informatie din cont, creez obiecte JSON si le introduc intr-un fisier JSON.

2. Grid:

Clasa folosita pentru generarea si modelarea mapei.

Metode folosite:

printMap:

Aceasta metoda printeaza mapa de joc folosindu-se de informatiile celulelor (tipurile acestora) si de starea de vizitare a acestora.

generateMap:

Aceasta metoda genereaza o mapa random, folosindu-se de exclusiuni (celulele generate deja nu vor mai fi folosite -> pozitiile lor vor ramane intacte).

goEast, goWest, goNorth, goSouth:

Metode folosite pentru deplasarea pe mapa. Dupa deplasare, atributul isVisited va fi setat corespunzator.

3. Account:

Clasa folosita pentru crearea si modelarea conturilor jucatorilor.

In cazul clasei Account, clasele Information si InformationBuilder folosesc agregarea pentru a spori incapsularea. Clasa Information este folosita pentru modelarea informatiilor despre jucator si implementeaza design pattern-ul Builder (InformationBuilder creaza noi obiecte de tip Information).

4. Entity

Clasa folosita pentru modelarea entitatilor (caractere si inamici).

Aceasta clasa implementeaza Element, interfata folosita pentru a implementa combat-ul cu Spell-uri dupa design pattern-ul Visitor.

5. Character:

Clasa folosita pentru a modela tipuri de caractere. Aceste caractere se vor crea cu ajutorul clasei CharacterFactory (se foloseste design pattern-ul Factory). La instantiere se vor crea random diferite spell-uri pentru caracter.

6. Spell

Clasa folosita pentru modelarea diferitelor tipuri de Spell. Aceasta clasa este folosita de asemenea pentru implementarea combat-ului cu Spell-uri dupa design pattern-ul Visitor. Astfel, spell-urile vor fi vizitatorii, iar entitatile vor fi "gazdele". "Gazdele" vor primi vizitatorii doar daca se indeplineste conditia ca gazda sa nu fie imuna la tipul vizitatorului.

7. Shop

Clasa folosita pentru modelarea shop-urilor. La instantiere se vor adauga in shop random diferite tipuri de potiuni.

8. firstPage

Clasa folosita pentru prima pagina a GUI. In aceasta pagina am implementat autentificarea, alegerea unui caracter (cu posibilitate de creare a noi caractere).

9. lastPage

Clasa folosita pentru ultima pagina a GUI. In aceasta pagina sunt afisate informatii despre caracterul curent, dupa finalizarea jocului.

Bonusuri:

Salvarea progresului in fisiere JSON

Crearea de caractere noi prin intermediul GUI

Randomizarea mapei folosind formule si excluziuni