Tema 1 - LOOP - League of OOP

Obiective

- familiarizarea cu Java și conceptele de bază ale POO
- fundamentarea practică a constructorilor și a agregării/moștenirii
- dezvoltarea unor abilități de bază de organizare și design orientat-obiect
- respectarea unui stil de codare si de comentare

Scenariu

Suntem într-un joc MMO-style. Eroii noștri își petrec viața într-un univers 2D, pe care îl explorează și în care se dezvoltă.

După cum este cunoscut din jocurile MMO și RPG, personajele noastre sunt diverse. Fiecare aparține unei anumite clase și are un anumit set de abilități la care se pricepe cel mai bine și pe care le poate dezvolta cel mai repede; le vom analiza în detaliu mai jos.

La începutul jocului, eroii noștri sunt plasați pe hartă în locuri bine definite. Apoi vor exista runde de durată unitate, în care toți eroii vor executa câte o anumită mișcare (bine definită) pe hartă.

Când doi eroi ajung în același loc, ei se vor lupta. În această rundă, fiecare erou combatant își va folosi toate abilitățile disponibile împotriva adversarului, o singură dată. După luptă, ei își vor vedea de drum începând de runda următoare.

La sfârșitul jocului (un număr stabilit de runde, cu locații inițiale și mișcări stabilite), programul nostru se va uita la eroii rămași în viață.

Harta

Harta pe care se desfășoară jocul este un dreptunghi 2D, compus din locații (“pătrățele”) de dimensiune unitate. Fiecare locație are un anumit set de proprietăți; tipurile de locații sunt:

- Land
- Volcanic
- Desert
- Woods

Eroi

Tipurile de eroi disponibile în LOOP sunt:

- Knight
- Pyromancer
- Rogue
- Wizard

Toți dispun de un număr de hit points (HP, also known as “viață”) și de un număr de puncte de experiență (XP). Există, de asemenea, și un mecanism de level-up în funcție de experiența câștigată.

Experiență și level up

Toate personajele au **XP iniţial = 0**, corespunzător nivelului 0. La nivelul 0 toate abilităţile au doar efectul de bază. În momentul în care un personaj câştigă o luptă (își omoară adversarul) XP-ul său va creşte după următoarea formulă :

```
XP_Winner = Xp_Winner + max(0, 200 - (Level_winner - Level_loser) * 40)
```

După cum se poate observa şi din formulă, câştigarea unei lupte cu un adversar care este cu cel puţin 5 nivele mai mic nu va aduce nici un plus de XP, dar câștigul în fața unui oponent de nivel mai mare poate aduce chiar mai mult de 200XP.

Pragurile de XP necesare pentru a face level-up se vor calcula după formula :

```
XP_level_up = 250 + nivel_curent * 50
```

La level up un erou va reveni la 100% HP.

Abilități

Fiecare tip de erou are un anumit set de abilități, ai căror parametri sau ale căror efecte depind de **terenul pe care se desfășoară lupta (land modifiers) și de personajul asupra căruia acționează (race modifier)**. Detaliem în continuare.

Pyromancer

Abil în manevrarea focului.

HP: 500 initial, +50 per nivel.

Fireblast - damage mare în runda curentă.
Damage: 350, +50/level

Victima	Modificator
Rogue	-20%
Knight	+20%
Pyromancer	-10%
Wizard	+5%

Ignite - damage în runda curentă + damage mai mic în umătoarele 2 runde.

- base damage: 150, +20/level
- 50 damage per runda, +30/level.

Modificatorii de mai jos se aplică atât pentru base damage, cât și pentru damage periodic:

Victima	Modificator
Rogue	-20%
Knight	+20%
Pyromancer	-10%
Wizard	+5%

Pe terenul de tip Volcanic Pyromancer-ul se hrăneşte cu energia mediului înconjurător iar abilităţile sale dau cu 25% mai mult damage.

Knight

HP: 900 initial, +80/level.

Execute - damage în runda curentă; dacă adversarul are un număr de HP mai mic decât o anumită limită, va fi ucis instantaneu.

- base damage: 200, +30/level
- HP limit: 20% * viața teoretic maximă a victimei la nivelul ei; +1% /level, până la un maxim de 40%

Damage-ul (nu și limita de viață) se modifică în funcție de victimă ca mai jos:

Victima	Modificator
Rogue	+15%
Knight	+0%
Pyromancer	+10%
Wizard	-20%

Slam - damage + incapacitarea adversarului (imposibilitate de mișcare) pentru următoarea rundă.

- base damage: 100, +40 /level

Victima	Modificator
Rogue	-20%
Knight	+20%
Pyromancer	-10%
Wizard	+5%

Knight-ul este expert în lupta corp la corp. Pe terenul de tip land el este mai puternic decât celelalte clase fiind specializat pe lupta în câmp deschis și primește 15% bonus damage.

Wizard

HP: 400 initial, +30 per nivel.

Wizard-ul are o capacitate mentală superioară care îi permite să reziste în mediul de deșert prin meditație și imbunătățirea abilităților. Pe terenul de tip **Desert** abilitățile sale ofensive sunt cu **10%** mai puternice.

Drain - scade din viața adversarului proporțional cu cât are deja.

- procent: 20%, +5% /level
- HP de bază: `min(0.3 * OPPONENT_MAX_HP, OPPONENT_CURRENT_HP)`
- ⇒ `damage = procent * min(0.3 * OPPONENT_MAX_HP, OPPONENT_CURRENT_HP)`

Modificatorii de mai jos se aplică asupra variabilei `procent`. Ei sunt multiplicativi, i.e. dacă de exemplu avem un Wizard level 4 (⇒ 40%) și el aplică Drain asupra unui Rogue, avem un procent total de 40% * 0.8 = 32%, deci formula de calcul de damage devine `damage = 0.32 * min(0.3 * OPPONENT_MAX_HP, OPPONENT_CURRENT_HP)`.

Victima	Modificator
Rogue	-20%
Knight	+20%
Pyromancer	-10%
Wizard	+5%

Deflect - dă damage egal cu un procent **din damage-ul total (*fără race modifiers*) pe care îl primește de la adversar**

- procent: 35%, +2% /level, până la un maxim de 70%
- *nu are niciun efect asupra unui Wizard* (doi eroi de tip Wizard nu își dau reciprc/recursiv damage)

Victima	Modificator
Rogue	+20%
Knight	+40%
Pyromancer	+30%
Wizard	N/A

Rogue

Expert în sneak-attacks.

HP : 600 initial, +40 /level

Backstab - damage în runda curentă, cu posibilitate de critical hit.

- base damage: 200, +20 dmg /level.
- O dată la 3 lovituri (lovitură = aplicat Backstab, pe orice teren) Rogue-ul poate da **1.5x** damage, **doar dacă în acel moment se afla pe terenul de tip Woods**, altfel se reia ciclul.

Victima	Modificator
Rogue	+20%
Knight	-10%
Pyromancer	+25%
Wizard	+25%

Paralysis - damage prelungit + incapacitarea adversarului pentru un număr de runde

- damage/rundă: 40, +10/level - se aplică în runda curentă (în care se desfășoară lupta) + rundele extra
- număr de runde overtime: 3 (6, dacă lupta se desfășoară pe teren Woods)

Victima	Modificator damage
Rogue	-10%
Knight	-20%
Pyromancer	+20%
Wizard	+25%

Datorită abilităților sale de camuflaj Rogue-ul este mai puternic pe terenul de tip pădure (**Woods**). Pe acest tip de teren Rogue-ul primește **15%** bonus damage.

Mecanism de joc

Jocul este bazat pe runde.

Într-o rundă toate personajele execută câte o mișcare (bine determinată), iar dacă două ajung în aceeași locație, se luptă. O luptă funcționează astfel: fiecare erou își va calcula parametrii propriilor abilități, în funcție de nivelul lor și de terenul pe care se află. Apoi fiecare abilitate va fi modificată în funcție de victimă. După aplicarea abilităților, eroii își vor calcula noile HP și XP după caz, după care runda se încheie.

Mențiuni

- Nu vor exista cazuri în care să se lupte mai mult de două personaje în același loc.
- La abilitățile cu efect prelungit (pe mai multe runde), dacă un personaj deja se află sub efectul unei abilități prelungite și suferă de la o nouă abilitate cu efect prelungit, noul efect îl va înlocui pe cel vechi; dacă de exemplu un erou mai are de suferit 5 runde de pe urma unui Paralysis și un Warrior îi aplică un Slam, atunci imobilizarea lui va dura doar o rundă și nu va mai primi damage pe rundele următoare.
- Damage-ul se calculează cu rotunjire la `int`; folosiți `Math.round`.

Cerințe

Ne dorim să modelăm acest joc în stilul orientat-obiect. Vom citi configurația și desfășurarea unui joc dintr-un fișier de intrare, vom rula jocul și vom scrie într-un fișier de ieșire stările eroilor noștri.

În soluțiile voastre, entry-point-ul (metoda `public static void main(String[] args)`) va fi într-o clasă numită `Main` dintr-un pachet numit `main`. Primul argument este numele fișierului de intrare, al doilea este numele fișierului de ieșire. Nu schimbați numele clasei Main, numele pachetului main sau ordinea argumentelor.

Input

Pe prima linie a fișierului se află 2 numere N si M reprezentând dimensiunile terenului. Pe următoarele N linii se vor găsi N string-uri de lungime M fiecare, cu litere corespunzătoare numelor terenurilor (W,L,V,D).

Pe următoarea linie se va gasi un număr P reprezentând numărul de personaje. Pe următoarele P linii se vor găsi câte un string reprezentând rasa fiecărui personaj și poziția lui inițială (e.g. “W 0 1” înseamnă un Wizard poziționat pe rândul 0 coloana 1).

Pe următoarea linie se găsește R, numărul de runde. Fiecare rundă e descrisă de P litere din mulțimea 'U'(up ⇔ rând–), 'D'(down ⇔ rând++), 'L'(left ⇔ col–), 'R'(right ⇔ col++), sau '_' (fără mișcare) indicând direcția în care se mișcă fiecare personaj. Direcțiile sunt atribuite eroilor în ordinea în care au fost descrise personajele in fișierul de intrare.

Output

În fișierul de ieșire veți adăuga P linii în formatul `rasa_pers level_pers xp_pers hp_pers row col`, câte una pentru fiecare erou, în ordinea în care au fost "declarați" inițial în fișierul de intrare. Dacă personajul este mort atunci în loc de statistici, pe linia corespunzătoare lui puneți "dead" (e.g. `K dead` înseamnă un Knight mort).

API

Vă punem la dispoziție un API (aici API-ul compilat pentru pachetul main) care vă permite lucrul cu fișiere. `FileIO` este clasa pe care o veți instanția pentru fiecare interacțiune cu fișierele. Ea constă în:

- constructorul `FileIO(String filename, boolean read)`; `filename` este numele fișierului cu care interacționați, `read` este setat `true` dacă interacțiunea este de citire, `false` pentru scriere
- metoda `String readLine()` - întoarce următoarea linie preluată din fișier
- metoda `void writeLine(String line)` - adaugă o linie nouă în fișier
- metoda `close()` - închide fișierul

Pentru includerea acestei clase în proiect consultaţi secţiunea Utile, aflată mai jos.

De asemenea, metodele din clasa String (`split`, `charAt`, `length`, `toCharArray` etc) vă pot fi de mare ajutor. Consultați documentația clasei String (link în secțiunea Utile).