3. feladat - Dice Wars

A feladat leírása

A feladat egy egyszerű stratégiai játék megvalósítása Java nyelven. A játékot egy játékos és n gépi játékos játssza. A játék elején lehessen kiválasztani, hogy hány gépi ellenfelet szeretnénk (min. 1).

A játékot egy tetszőleges méretű pályán játsszák, amely N*M mezőből áll. Minél több játékos játssza a játékot, annál több mezőből érdemes felépíteni a játékteret. A pálya tartalmazhat semleges területeket, amelyek nem tartoznak egy játékoshoz sem, és nem lehet őket elfoglalni. A pálya formája tetszőleges lehet. A pálya lehet négyzetrács, illetve hatszögrács alapú. Négyszögrács esetében egy mezőnek 4, míg hatszögrács esetében 6 szomszédja van.

Kezdetben minden játékos egyenlő számú mezővel rendelkezik, amelyek véletlenszerűen kerülnek kiosztásra. A program minden lépés után jelenítse meg, hogy kinek hány mezője van. A mezőkön dobókockák helyezkednek el, minden mezőn minden esetben van legalább 1, maximum pedig 8 dobókocka. A játék indulásakor minden játékos háromszor annyi dobókockát kap, mint amennyi területe van (ha például 6 területtel kezd minden játékos, akkor 3*6 dobókockát kap mindenki). A dobókockákat a területek között véletlenszerűen kell kiosztani, viszont fontos, hogy minden területen legyen minimum 1 dobókocka. Egy területen legfeljebb 8 dobókocka lehet.

Miután megtörtént a pálya inicializálása, tehát minden játékos megkapta a területeit és a dobókockáit, elkezdődik a harc. Minden játékos a megadott sorrendben követi egymást, egyszerre csak egy játékos cselekszik. A cél az, hogy minden területet elfoglaljunk. Egy területet úgy lehet elfoglalni, hogy megtámadjuk egy mellette lévő mezővel. Csak olyan mezőről támadhatunk, amelyen legalább két dobókocka áll. Hiszen, ha elfoglaljuk az ellenség területét, a saját mezőn is maradnia kell egy dobókockának, ill. az ellenféltől elnyert területre is tennünk kell legalább egy dobókockát.

A harc kimenetele a következőképpen alakul: mindkét játékos a védekező és a támadó is annyi dobókockával dob, amennyi dobókocka az adott mezőn áll (ha pl. valaki 5 dobókockával dob, akkor a legnagyobb érték 5*6 = 30 lehet, amíg, ha az ellenfél pl. csak 1 dobókockával dob, akkor a legnagyobb érték, amit dobhat, az 6). A harcot az nyeri, aki nagyobb számot dob. A dobott értékeket jelenítsük is meg a képernyőn.

- Ha a támadó nyer, a védekező elveszíti az összes dobókockáját, ami a területen volt, a támadó összes dobókockája az elnyert mezőre vándorol, kivéve egyet, amelyet az eredeti mezőn hagy
- Ha a védekező nyer, a támadó elveszíti az összes dobókockáját kivéve egyet, amely a támadó mezőn marad.
- Amennyiben mindkét fél ugyanakkorát dobott, az esetben a védekező játékos nyer. (lásd előző pont)

Egy körben egy játékos annyiszor támadhat, amennyiszer akar, tetszőleges mezőkről. Amikor egy játékos már nem akar tovább támadni, átadhatja a kört a soron következő játékosnak. Amikor valaki a kör átadást választja, akkor a területei után k darab dobókockát kap

(k = birtokolt területek száma / 2). Ezeket a plusz dobókockákat véletlenszerűen kell szétosztani a játékos által birtokolt területek között. Miután ez megtörtént, a következő játékos jön.

Amennyiben egy játékosnak nincsen több területe kiesett a játékból. A játékot az a játékos nyeri, aki utoljára marad, és birtokolja az összes mezőt.

Valósítsunk meg több fajta gépi ellenfelet. Lehet támadó, védekező vagy bonyolultabb feltételrendszerek alapján döntő. Például lehet olyan, aki mindig támad, amikor csak tud, lehet aki csak akkor támad, ha bizonyos valószínűséggel győz, vagy aki figyelembe vesz több környező területet is.

A játék végeztével lehessen a teljes játék menetét fájlba menteni. Ez nyilván tartalmazza a játék kezdeti állapotát (melyik mező kihez tartozik, ott hány kocka van stb.). Emellett az összes ezután következő lépést és dobás eredményét. Tehát a fájl alapján a teljes játék menetét lehessen látni. Például egy támadás esetén amire szükségünk van, hogy a két játékos melyik kockával hányas értéket dobott. Eltárolhatjuk ezt is, de tetszőleges más formában is megvalósítható a mentés, a lényeg, hogy a teljes játék menetét tartalmazza, az adatokból vissza lehessen játszani a játékot.

Legyen lehetőség egy játékmenetet betölteni, és lépésenként visszajátszani. Ilyenkor a játékmenet ugyanaz, de az inputok nem a felhasználótól (illetve a géptől) jönnek, hanem a megadott játékmenet-fájl alapján történnek meg.

Javítási útmutató

Funkcionális követelmény

- Van lehetőség egyedül játszani, gépi ellenfelek ellen. A játék elején ki lehet választani, hogy hány ellenfelet szeretnénk - **5%**
- A játék indulásakor a pálya létrejön. Minél több ellenfél van, a pálya annál nagyobb. A játék elején minden játékos egyenlő számú mezővel rendelkezik (kivétel, ha nem osztható a területek száma a játékosok számával). A kockák véletlenszerűen vannak kezdetben kiosztva, de minden mezőn van legalább 1 db - **5%**
- A játékosok egymás után jönnek. Egy játékosnak tetszőleges számú lépése lehet. Minden lépés után meg van jelenítve, hogy kinek hány mezője van. - **5%**
- A mezőkön mindig van legalább 1 kocka, legfeljebb 8 kocka, ez a szabály a játék során soha nem bomlik meg. Minden kör végén az adott játékos megfelelő mennyiségű dobókockát kap, a területei között véletlenszerűen kisorsolva - **5%**
- A játékos meg tudja támadni a szomszédos területeket a legalább 2 kockával rendelkező saját területeivel. Ha 1 kockánk van, akkor nem támadhatunk. Csak saját mezőről támadhatunk és csak ellenséges mezőt. A program ellenőrzi, hogy csak szomszédos mezőt tudjunk megtámadni, mást nem. A játékos azt is kiválaszthatja, hogy melyik mezőről indítja a támadást - **5%**
- Támadáskor a megfelelő mennyiségű kockákkal dobás történik. Az egyes kockák által dobott értékeket a program kijelzi, illetve az összeg is látszódik. Az nyer, akinek nagyobb a dobott értéke. A megadott szabályok alapján vagy a támadó veszti el a kockáit, vagy elfoglalja az ellenfél területét, arra 1 kivételével a többi kocka átkerül - **10%**
- Ha a játék során az összes területünket elfoglalják, akkor kiesünk a játékból. Ilyenkor a játék vagy végetér, vagy végignézhetjük, ahogy a gépi ellenfelek eldöntik az eredményt maguk között, a játékos beavatkozása nélkül - **5%**
- Meg van valósítva legalább 3, különbözően viselkedő gépi ellenfél. Az ellenfelek típusát vagy kiválaszthatjuk, vagy véletlenszerűen (vagy valami más alapján) választódnak - **5%**
- Van olyan ellenfél-típus, aminek működése nem triviális, azaz nem csak random lépked, hanem van esélye is a győzelemre - **5%**
- A teljes játékmenetet el lehet menteni fájlba. A fájl tartalmaz minden adatot, ami alapján a játékot vissza lehet (vagy lehetne) játszani - **5%**
- Egy mentett játékot vissza lehet tölteni. Ilyenkor a lépéseknek követhető módon jelennek meg. Pl. kis késleltetéssel az egyes lépések között / enter lenyomására érkezik a következő lépés - **5%**

Megvalósítási követelmény

- A program tartalmaz legalább 5 különböző, értelmes osztályt, amelyek egy-egy, a játékhoz kapcsolódó entitást írnak le - **5%**
- A programok az osztályok funkcionalitásuk alapján package-ekbe vannak rendezve - **5%**
- A programban szerepel legalább 2 olyan osztály, aminek van több, értelmes gyerekosztálya - **5%**

- Minden osztálynak 1 (egy) meghatározott feladata van, egy osztály nem végez több, lényegében különböző műveletet - **5%**
- Az osztályok, adattagok és metódusok nevei konvenció szerint vannak elnevezve, a nevek tükrözik az osztályok, metódusok, adattagok feladatait. A program megfelelő módon van indentálva - **5%**
- Nincsenek a programban hosszú metódusok. A programban sehol nincs 100 sornál hosszabb metódus, és a metódusok nagy része kevesebb, mint 50 sorból áll. Ezek alól kivételt képeznek a GUI-val rendelkező programok esetén a grafikus elemek létrehozására, megjelenítésére szolgáló metódusok - **5%**
- A program JavaDoc dokumentációval el van látva. Az osztályok mindegyike meg van magyarázva JavaDoc dokumentációval, és a metódusoknak is legalább a 70%-a (a getter és setter metódusokon kívül) - **5%**
- A program fel van készülve érvénytelen felhasználói inputokra. Ezek vagy meg vannak tiltva, vagy le vannak kezelve. Pl. nem létezik a megadott fájl, olyan cselekvést szeretnénk végrehajtani, ami az adott helyzetben nem megengedett, szám helyett szöveget adunk meg, nem a megfelelő értéktartományból írunk be értékeket, stb. Az ilyen esetek nagy része le van kezelve - **5%**