

Aknakereső

Programozás 3 házi feladat

Háromi Bence

AK51GI

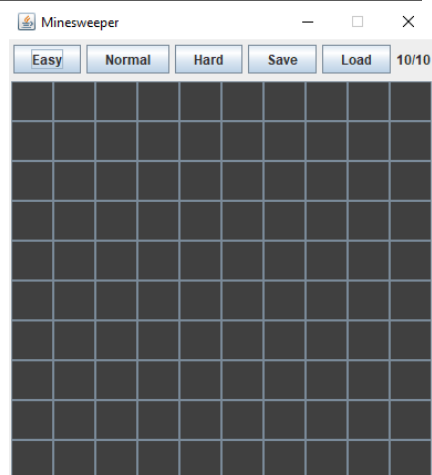
LEÍRÁS

A játék kezdetekor egy letakart táblát látunk, ami cellákból épül fel, ezek aknákat rejthetnek. A cellákat az egér bal gombjával kattintva felfedhetjük, így látjuk, hogy mi van benne. Ha szerencsénk volt akkor üresre kattintottunk és folytathatjuk a játékot, ha viszont aknát sikerült felfednünk, akkor „felrobbanunk” vagyis vége a játéknak. Ha üres cellára kattintottunk, és nincsen a közvetlen közelében akna, akkor elénk tárul a körülötte lévő aknamentes terület. Ha egy akna közvetlen szomszédságába kattintunk, akkor csak az az egy cella nyílik meg amelyikre kattintottunk benne egy számmal, ami azt mutatja meg, hogy hány akna található a mellette lévő cellákban (1-8 db). Lehetőségünk van az egér jobb gombjával zászlókkal megjelölni, az általunk gondolt aknarejtő cellákat, természetesen ezek eltávolíthatók a zászlóra újból jobb egérrel kattintva. A játék kezdetekor megtudjuk, hogy mennyi akna található a pályán és célunk ezeket kikerülve, minden aknamentes cella felfedése, amint ez megtörtént nyertünk.

A játékosnak a játék indításakor lehetősége van új játékot kezdeni, ranglistát nézni, illetve kilépni. A játék kezdetekor egy óra indul, mely méri mennyi idő alatt sikerül a játékot megnyerni, ez alapján rangsorolódnak a nyertesek.

FELHASZNÁLÓI KÉZIKÖNYV

A játék elindításakor a 10x10-es aknamezőt láthatjuk magunk előtt (a jobb oldali képernyőkép szemlélteti), ami alap esetben könnyű (Easy) nehézségű, vagyis 10 akna van elrejtve benne. A többi nehézségi fokozat, közepes (Normal) és nehéz (Hard), ezek 20, illetve 30-at rejtenek. A nehézséget a felső sorban lévő gombokkal választhatjuk ki, ha megnyomjuk valamelyiket automatikusan elindul a játék és kezdhethetjük is az aknavadászatot.



Cellát felfedni a bal egér, zászlót elhelyezni (megjelölni) a jobb egér segítségével tudunk. Zászlót eltávolítani is van lehetőségünk, egy zászlós cellát a jobb egérgombbal megnyomva. Ha elhelyezünk egy zászlót, akkor a jobb felső sarokban lévő aknaszámláló értéke eggyel csökken, ezzel segítve, minket, hogy még hány aknát kell megtalálnunk.

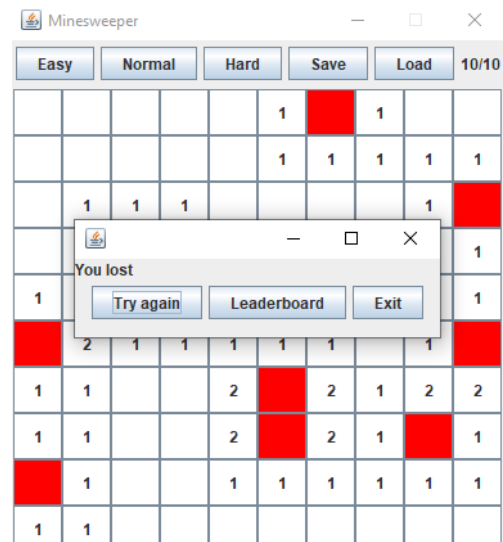
Ezen kívül lehetőségünk van még menteni a játékállást a save gomb segítségével, ha később folytatni szeretnénk. Betölteni a load gombbal tudjuk az utoljára elmentett játékállást.

Ha aknára kattintottunk, akkor ez a képernyő fogad minket:

Itt lehetőség van újratekinteni a játékot a Try again gombbal, megtekinteni a ranglistát a Leaderboard gombbal, illetve kilépni az Exit gombbal.

Ha megnyertük a játékot, ugyanezt az ablakot láthatjuk magunk előtt, csak You win felirattal. Ekkor a leaderboard gomb megnyomásakor egy felugró ablakba beírva nevünket, felkerülünk a ranglistára, itt láthatjuk a többi nyertes nevét, idejét másodpercben illetve, hogy hol helyezkedünk el a többiekhez képest.

E rövid ismertető után kezdődhet a játék! Jó szórakozást!



STRUKTURÁLIS LEÍRÁS

AZ OSZTÁLYOK LEÍRÁSA

Cell

Cella, mely tárolja, hogy tartalmaz-e bombát, fedetlen-e, mennyi az értéke, illetve megjelölt-e.

Függvényeivel a változóinak értékeit kérdezhetjük le, illetve állíthatjuk be.

Serializálható, hogy menthető legyen.

Table

Létrehozza és később ellenőrzi a táblát, illetve kimenti/betölti a tartalmát fájlba serializálással.

Egy 2d-s ArrayList-ben tárolja a cellákat, ezen kívül még a tábla sorainak, oszlopainak, illetve a fedett cellák számát tárolja.

Fontosabb függvényei:

- `setMines(num_of_bombs: int): void`
elhelyezi a megadott számú aknát véletlenszerűen a táblán
- `calculateValues(): void`
kiszámolja az üres cellák értékeit, ami megegyezik az adott cellával szomszédos aknát tartalmazó cellák számával.
- `revealAll(): void`
felfedi az összes cellát
- `discover(x: int, y: int): int`
felfedi az aknamentes területet adott x sorú és y oszlopú cellából indulva és visszaadja a még fedett cellák számát. Ha értéke 0, akkor felfedi a szomszédos aknákat is, ha nem akkor csak a megadott egyet.
- `getNumOfMines(): int`
megszámolja és visszaadja az aknák számát a táblában.
- `getNumOfMarked(): int`
megszámolja és visszaadja a megjelölt cellák számát a táblában.

Game

Tárol egy táblát, aknaszámot, fedett cellák számát, a játék kezdetekori időt, illetve a játékos idejét.

Fontosabb függvényei:

- `createGame(): void`

létrehoz egy játékot, vagyis elhelyezi az aknákat, illetve kiszámolja a cellák értékeit

- setTime(): void
beállítja a játékos idejét: a játék befejezési idejét kivonja a játék kezdőidőpontból az időt ezzel megkapva a játékidőt.
- discover(x: int, y: int): boolean
a tábla azonos nevű függvényét hívja meg. Ha a kapott fedett cellák száma megegyezik a az aknaszámmal, akkor igazat ad vissza, különben hamisat.

Player

A játékos nevét, illetve idejét (másodpercben) tárolja.

Serializálható, a mentés érdekében.

Leaderboard

Egy arraylistben tárolja a játékosokat.

Fontosabb függvényei:

- saveToFile(): void
a listát kimentti egy fájlba.
- loadFromFile(): void
fájlból betölti a listát.
- addPlayer(name: int, time: int):void
- hozzáad a listához egy játékost, majd rendezi a listát az idő szerinti növekvő sorrendbe.

Display

A játék grafikája, Swinggel megvalósítva.

Tesztelés

A tesztelés JUnit 4 segítségével történik, 3 osztály (Game, LeaderBoard, Table) összesen 10 metódusának tesztelése.

OSZTÁLYDIAGRAM

