

# Dokumentáció

## Aknakereső

### Játék leírása

Az aknakereső játék egy  $n \times m$ -es táblán játszható. Ezen a táblán minden egyes mezőn vagy van egy akna vagy nincsen semmi. Az a feladat, hogy megtaláljuk az összes aknát. Ehhez fel lehet forgatni minden mezőt, viszont ha olyat fordítunk meg amelyiken akna van akkor veszítettünk és vége a játéknak. Abban az esetben ha nincs akna a mező akkor felfordul a mező. A felfordult mezőn megjelenik a mező egy szám, ami a szomszédos aknák számát mutatja (ha nincs akna a szomszédságban akkor üres marad). Továbbá, ha a felfordított mezőnek nincsen akna szomszédja, akkor felfordítja az összes szomszédját, amik ugyanúgy viselkednek mintha mi forgattuk volna őket meg. Ennek következtében egy számokkal határolt területet kapunk aminek a belseje aknamentes. A játéknak akkor van vége, ha minden aknamentes mezőt felforgattunk. A játékban 3 szint van és az számít, hogy milyen gyorsan sikerült megtalálni az aknákat. A legjobb 10 eredmény eltárolásra kerül minden kategóriában. Minden játék után megjelenik a ranglista és ha kell módosul is. Azokat a mezőket ahol úgy sejtjük akna van meg tudjuk jelölni, így nem lehet felfordítani őket. Egy számláló jelöli, hogy még mennyi akna van hátra (azokat számolja amiket megjelöltünk attól függetlenül, hogy valójában van-e akna vagy nincs).

### Use-Case-ek

- **Kiválasztani a nehézségi szintet (Easy, Medium, Hard):** A játék indítását követően legördülő menüből tudunk választani.
- **Elkezdeni játszani:** A játékot az első mezőre történő kattintás indítja el (megkezdődik az időmérés).
- **Újrakezdeni a játékot:** Egy megkezdett kört újra lehet indítani egy annak dedikált gombbal. Megszakításra nincsen lehetőség.
- **Mezőt felfordítani:** Bal klikkel a mezőre felfordul a mező.
- **Aknát megjelölni:** Jobb klikkel a mezőre aknának jelöli azt.

### Megoldás vázlata

A táblát a mezőkkel egy mátrixban tárolom el. Az egyes mezőket egy-egy osztály fogja reprezentálni aminek argumentuma, hogy van akna rajta vagy nincs, hogy fel van fordítva vagy sem illetve, hogy mennyi akna van a szomszédságában. Az aknák az első kattintást követően lesznek véletlenszerűen szétszórva. Az egyes mezőkön végrehajtható

függvények a mező osztály függvényei. Van egy külön osztálya a táblának ami a táblán végrehajtható műveletekért felelős. Egy osztály felel a játék működéséért is. A ranglista elmentése szerializálás segítségével fog történni, ez minden egyes az eddiginél jobb eredményeknél frissül. A GUI-hoz Java Swinget fogok használni. Lesz benn tesztelés JUnit segítségével.

## Felhasználói kézikönyv

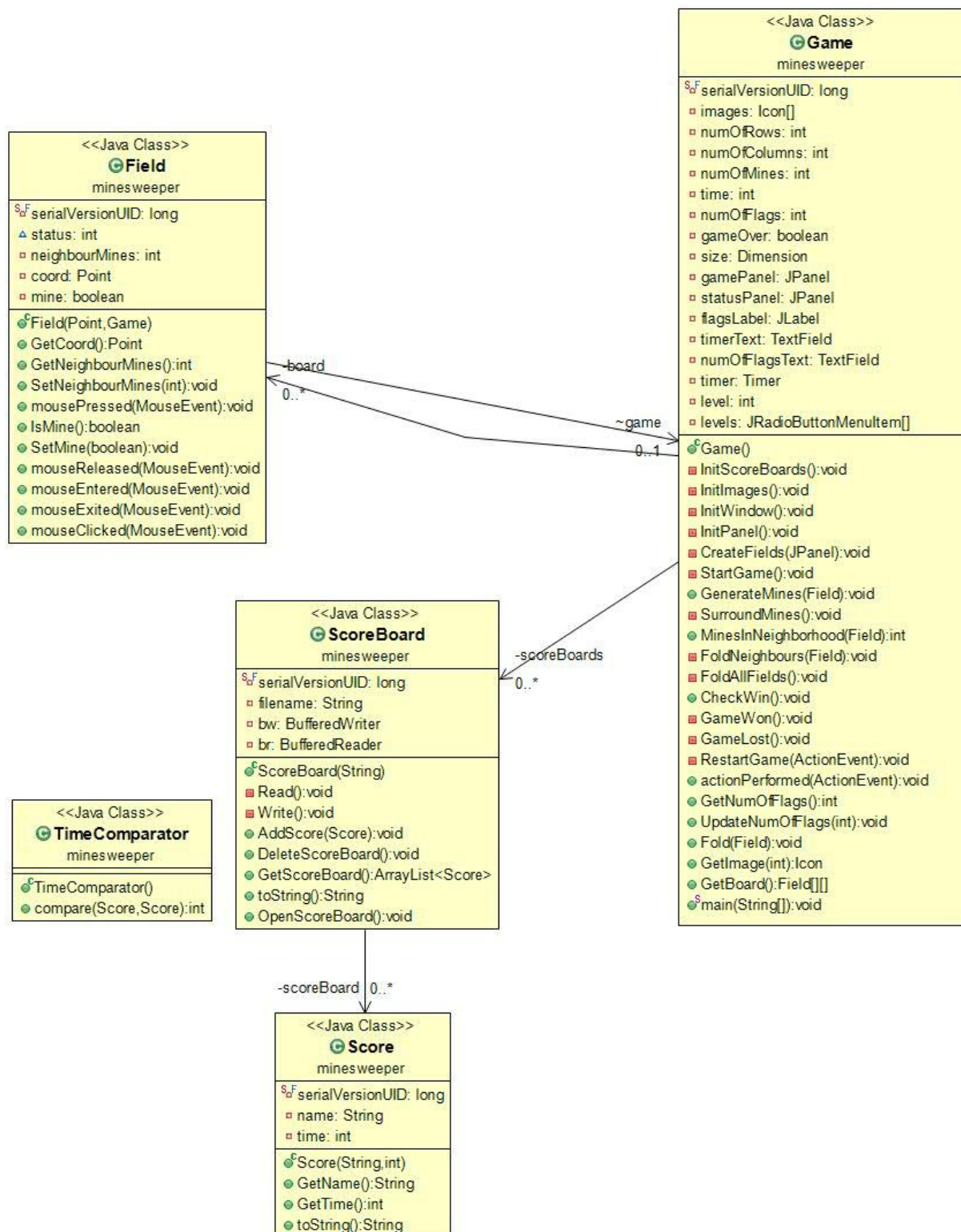
A játék elindulása után alaptól kezdő szinten lehet játszani (10\*10-es pálya 10 aknával). A játékosnak lehetősége van a nehézségi szint megváltoztatására (haladó: 16\*16-os pálya 40 aknával, profi: 16\*30-as pálya 99 aknával) a felső menüsávon található Level megnyomását követően. Ha új szintet választ a játékos eldöntheti, hogy újra akarja-e kezdeni a játékot vagy sem (csak úgy válthat nehézséget, ha újrakezdi).

Ha elkezdi a játékot akkor a mezőkre bal gombbal való kattintással azokat felfordítja a játékszabályoknak megfelelően. Ha rákattint jobb gombbal egy mezőre akkor azt megjelöli aknaként (nem lehet felfordítani), ha még egyszer rákattint jobb gombbal akkor kérdőjellé teszi, ami azt jelenti, hogy nem tudja eldönteni akna vagy nem akna a mező (ha még egyszer jobbklikkel a mezőre akkor megjelöletlenné teszi).

Ha olyan mezőre kattint a játékos ahol akna van, akkor vége a játéknak, ekkor eldöntheti játékos, hogy ki szeretne lépni, vagy újra szeretné kezdeni a játékot.

Ha sikeresen megnyerte a játékos a játékot akkor az ideje alapján hozzáadja a dicsőségtáblához, ami az első tíz eredményt tárolja szintenként. A táblát megmutatja a játékosnak. A játékos nevét az oprencertől kérdezi le. A nyereség után eldöntheti a játékos, hogy újra akar játszani vagy kilép.

# Osztálydiagram



# Megoldás

## Field osztály

Az osztály az egyes mezőket jelenti. Leszármazik a JButton osztályból és implementálja a MouseListener-t.

Attribútumai:

- private Game game - A játék, amihez tartoznak a mezők
- private int status - Azt mondja meg, hogy milyen állapotban van a mező (semmi, zászló, kérdőjel)
- private int neighbourMines - A szomszédos aknák száma
- private Point coord - A mező koordinátája a táblán
- private boolean mine - Van-e akna a mezőn

Metódusai:

- Konstruktor - Egy koordináta és egy Game alapján létrehozza mezőt. A mine-t false-ra, a status és a neighbourMines változókat 0-ra állítja
- Getter és Setter függvények - A változók beolvasására és lekérdezésére kellenek
- mousePressed - A mezőre kattintást kezeli le. Jobb kattintás esetén változtatja a státuszt, bal esetén felfordítja a mezőt és a módosításokat a game-be is átülteti

## Score osztály

Az osztály egy idő és egy név eltárolására alkalmas. Ezek alkotják a dicsőségtáblát.

Attribútumai:

- private String name - A játékos oprendszertől lekérdezett neve
- private int time - A játékos ideje

Metódusai:

- Konstruktor - Név és idő alapján létrehoz egy Score-t
- Getter és Setter függvények - A változók beolvasására és lekérdezésére kellenek
- toString - Átalakítja szöveggé

## ScoreBoard osztály

Eltárolja az első tíz eredményt egy kategóriában.

Attribútumai:

- private ArrayList<Score> scoreBoard - Az eredmények listája
- private String filename - A file neve amiben tárolja az eredményeket
- private BufferedWriter bw - Fileba íráshoz kell
- private BufferedReader br - Fileből olvasáshoz kell

Metódusai:

- Konstruktor - Beolvassa paraméterül kapott fileból az eddigi top eredményeket

- Getter és Setter függvények - A változók beolvasására és lekérdezésére kellenek
- toString - Átalakítja szöveggé
- Read - Beolvassa fileból az Score-okat
- Write - Kiírja fileba a Score-okat
- AddScore - Hozzáad egy Score-t az eddigiekhez, ha kevesebb az ideje mint az eddigi legrosszabb akkor hozzáadja helyette és rendezi idő szerint
- DeleteScoreBoard - Törli a lista tartalmát, Unit teszthez kell
- OpenScoreBoard - Megnyitja egy JTable-ben a top tíz eredményt

## TimeComparator osztály

Idő alapú összehasonlításához használok a Score típusoknak. Implementálja a Comparator<Score>-t.

Metódusai:

- compare - Összehasonlít két Score példányt az idejük alapján és visszatér az eredménnyel

## Game osztály

A játék futásáért felel.

Attribútumai:

- private Icon[] images - A használandó képeket tárolja
- private ScoreBoard[] scoreBoards - A különböző szintekhez tartozó scoreBoard-ok
- private int numofRows - Tábla sorainak száma
- private int numofColumns - Tábla oszlopainak száma
- private int numofMines - Aknák száma
- private int time - Az idő a játék kezdete óta
- private int numofFlags - A lerakott zászlók
- private boolean gameOver - Vége van-e a játéknak
- private Dimension size - Ablak mérete
- private JPanel gamePanel - Játék panelja
- private JPanel statusPanel - Statuszok panelja
- private TextField timerText - Az időt mutatja
- private TextField numofFlagsText - Zászlók számát mutatja
- private Timer timer - A számlálást intézi
- private int level - A játék nehézségi szintje
- private JRadioButtonMenuItem[] levels - A szintek választásához tartozó gombok
- private Field[][] board - A mezőket tartalmazó mátrix

Metódusai:

- Konstruktor - Meghívja az inicializáló függvényeket és elindítja a játékot
- Inicializáló függvények - Beállítják a default értékeket a játék kezdetén, létrehozzák a pályát és a Swing elemeket
- Getter és Setter függvények - A változók beolvasására és lekérdezésére kellenek

- GenerateMines - Véletlenszerűen kiválaszt annyi mezőt ahány akna van (a paraméterül kapottn kívül) és ezeknél beállítja, hogy van rajtuk akna
- SurroundMines - Beállítja minden mezőnek az őt körülvevő aknák számát
- FoldNeighbourhood - Felfordítja az összes szomszédos mezőjét a paraméterül kapott mezőnek
- FoldAllFields - Felfedi az összes mezőt, a játék végén kell
- CheckWin - Visszatér, hogy nyert-e a játékos vagy sem
- GameWon - Meghívódik, ha nyert a játékos
- GameLost - Meghívódik, ha veszített a játékos
- RestartGame - A szintnek megfelelően új játékot kezd, újra beállítja a default értékeket
- actionPerformed - Meghívja a RestartGame függvényt, új szint választásakor hívódik meg
- UpdateNumOfFlags - A lerakott zászlók szerint változtatja a hozzájuk tartozó értéket
- Fold - Felfordítja a paraméterül kapott mezőt. Ellenőrzi vége van-e a játéknak és ha kell felfordítja a szomszédos mezőket is. Valamint ha ez az első kattintás a játékban elindítja a timer-t
- main - Létrehoz egy Game példányt ami elvégzi a játékot