

# Csapó Botond

## Aknakereső - Házi Feladat dokumentáció

### Osztályok dokumentációja

#### 1. Aknakereső osztály

Ez az osztály hozza létre a pályát, amelyen majd játszani lehet.

**újra** (JButton): Ekkor a mezők visszaállnak eredeti állapotukba (Enabled), és újra kiosztja a bombákat a program.

**zászlók** (JButton): Erre a gombra kattintva a játékos zászlókkal tudja megjelölni a játék mezőit.

**kattintás** (JButton): Ekkor a játékos kattintással felfedi a mezőt. (Alapvetően ez van beállítva játék indításakor)

**időmérő** (JTextField): Kiírja a játék kezdetétől eltelt időt.

**indikator** (JTextField): Tudatja a játékoskal, hogy most épp zászlókat tud e rakni, vagy kattintgatni tud.

**mezők** (JButton[][]): A játék mezői.

**számok** (int[][]): Az egyes mezők, a játékhoz szükséges információit tartalmazza, mint például ha bomba, akkor 10-es érték van benne, egyébként a szomszédos mezőkön található bombák száma.

**grid** (Container): Ebben vannak a mezők elhelyezve.

**a, b** (int): A mezők, számok hosszát, szélességét tartalmazó számok.

**katt** (boolean): Igaz az értéke, ha a kattintás gombot megnyomták, és hamis, ha a zászlókat. Értékétől függően máshogy viselkednek a mezők, amikor megnyomják őket.

**megy** (boolean): Az idő mérését segíti. Ha hamis akkor leáll az „idő”.

**idő** (Timer): A játék kezdete óta eltelt időt méri, és frissíti az „időmérő”-t.

**mérő** (int): Ezt az értéket növeli az „idő”, másodpercenként.

**teljesítmény** (String): A játékos idejét tárolom.

**főmenü** (JMenuItem): Ha a játékos nyert, akkor a program frissíti az őt elindító főmenü toplistáját.

#### 2. Főmenü osztály

Ez az osztály felel a főmenü létrehozásáért

**Játék** (JButton): Ennek a megnyomásakor elindul a játék.

**név** (JTextField): A játékos ide tudja beírni a nevét, amivel a toplistán szerepelhet majd.

**toplista** (Toplista): A főmenün megjelenített toplistát kezeli. Nem sima JTextField, mert az új adatokkal még kell dolgozni (rendezés, fájlba írás), ezért erre egy külön osztályt hoztam létre.

**fokok** (Object[]): Stringekből álló tömb. Ezt adom hozzá a JComboBoxhoz, így téve lehetővé a játék nehézségi fokozatának a kiválasztását.

**fokozatok** (JComboBox): Ezzel lehet kiválasztani a játék nehézségi szintjét.

**aknakereső** (Aknakereso): Az aknakereső, amit létrehoz a főmenü.

**játékosneve** (String): Ebben tárolom a játékos nevét, amit megadott a név mezőben.

### 3. Toplista osztály

Ez az osztály felel a főmenü toplistájának beállítására.

**főmenü** (Fomenu): Az a főmenü, amelyiknek a toplistáját frissíteni kell.

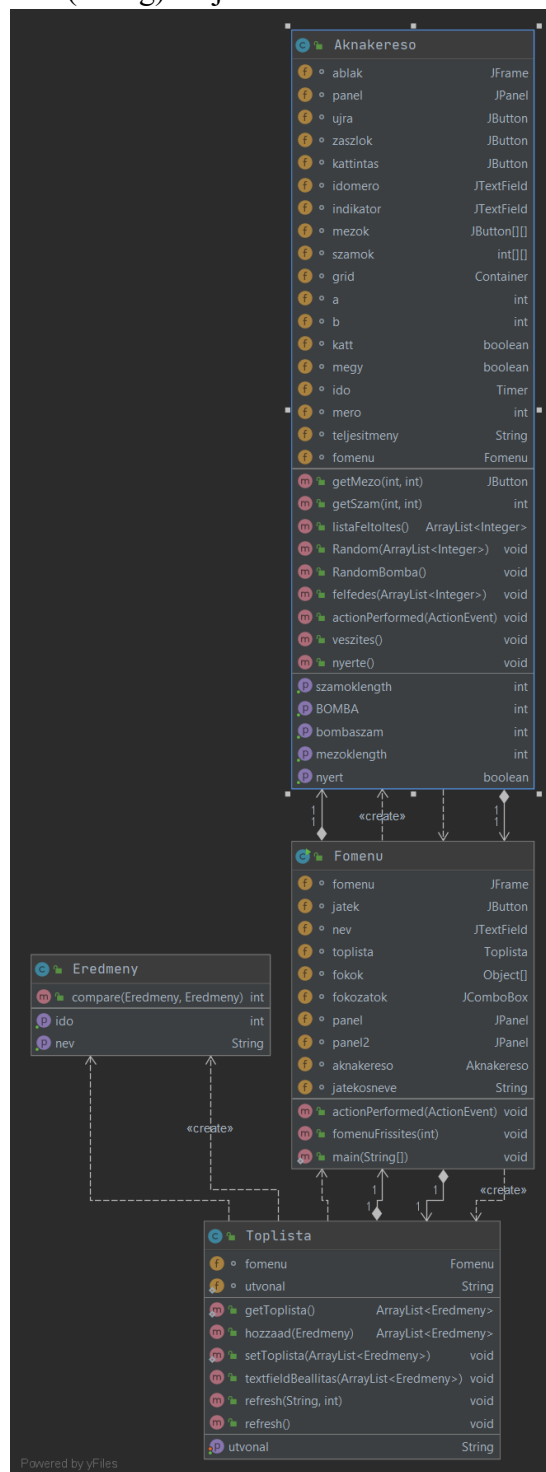
**útvonal** (String): A fájl elérési útvonalát tárolja.

### 4. Eredmény osztály

Tárolja az egyes játékosok eredményét, és össze tudja hasonlítani őket, idő szerint.

**idő** (int): A játékos elért ideje

**név** (String): A játékos neve



Az osztályok UML-diagrammja

## Metódusok rövid leírása

### 5. Az Aknakereső osztályban

**Konstruktor:** Beállítja a paraméterként kapott értékeket, elindítja az időmérőt, valamint beállítja azt a JFrame-et, amin a játék látszani fog. Mint például: panelek hozzáadása, gombok hozzáadása a panelekhez, stb. Illetve a Container feltöltése a JButton-okkal is még egy fontos feladata. Valamint meghívja a RandomBomba() függvényt.

**listaFeltoltes():** Visszatér egy Integerekből álló ArrayList-tel, amiben az egyes számok ezres és százasként helyiértékű elemei jelentik az X koordinátát a mátrixban, az egyes és tizedes helyiértékű számok pedig az Y-t. Az egyes elemeket így egy lineáris listába helyeztem el, amiben a (12, 3)-as koordinátájú mező 1203 ként jelenik meg a listában, és az X koordinátát úgy kapom meg, hogy ezt a számot osztom százal, az Y-t pedig úgy, hogy veszem a szám százasként modulóját.

**Random():** A koordináták információjával feltöltött listából véletlenszerűen választ bombákat, mégpedig úgy, hogy random kiválaszt egy számot a listából, majd annak a számnak a számok mátrixból megfeleltethető értékét tízzel teszi egyenlővé. Ezután eltávolítja a random kiválasztott számot, így kerülve el azt az esetet, hogy egy számot többször is kiválasszon.

**RandomBomba():** Meghívja a Random() és listaFeltoltes() metódusokat, illetve a maradék mezőkön megvizsgálja, hogy mennyi szomszéd van körülötte.

**felfedes():** Rekurzív függvény, amely akkor áll le ha már üres a paraméterként kapott ArrayList. Megvizsgálja minden irányban azt, hogy az ott található gombok engedélyezettek-e, illetve azt, hogy meg vannak-e zászlózva. Ha nincsenek engedélyezve, és nincs rajtuk zászló, akkor felfedi őket (kiírja értéküket, nem engedélyezi). Ha a mellette lévő mező szintén nulla, akkor arra is meghívja ezt a függvényt. Csak akkor hívódik meg, ha a mező, amire a játékos kattintott, nullás értékű (nincs körülötte bomba). Így elkerülve azt, hogy a bombákat is felfedje a program.

**actionPerformed():** Ha a játékos az újra gombot nyomja meg, akkor a pálya resetel, azaz új helyen lesznek a bombák, és megint engedélyezve lesznek a mezők. Ha a játékos a kattintás gombot nyomja meg, akkor amikor a mezőkre kattint, azok felfedik bomba voltukat, illetve ha nem azok, akkor a mellettük lévő bombák számát. Ha a játékos a zászlók gombot nyomja meg, akkor amennyiben a mezőkre kattint, azokon egy 'Z' betű jelenik meg.

**veszítés():** Ha a játékos bombára kattint, akkor hívódik meg a metódus. Leállítja az időmérőt, és felfedi a pálya összes elemét.

**nyerte():** Ha az összes mező fel van fedve, ami nem bomba, akkor a játékos nyer.

### 6. A főmenü osztályban

**Konstruktor:** Létrehozza az ablakot, frissíti a toplistát. Paraméterként azt az útvonalat kapja meg, ahonnan a toplistát tudja kiolvasni.

**actionPerformed():** Elmenti a játékos nevét, amennyiben ezt a játékos nem adta meg, „Névtelen”-re állítja. Ha a játékos a játék gombra kattint, akkor létrehoz egy aknakeresőt, aminek a mérete és a bombaszáma attól függ, hogy milyen fokozatot választott a játékos.

**fomenuFrissites():**Frissíti a toplistát.

**main():** Létrehoz egy új főmenüt.

## 7. A Toplista osztályban

Konstruktor: Létrehoz egy új JTextField-et, és beállítja a paraméterként kapott főmenüt.

getUtvonal()/setUtvonal(): Visszaadja, beállítja az útvonalat, ahonnan a szükséges adatokat meg tudja szerezni a Toplista.

getToplista(): Visszaad egy Eredményekből álló listát, amelyet a fájlból olvasott ki. (Beolvas sorokat, majd azokat feltördeli, az első darab lesz a név, a második pedig az elért idő)

hozzaad(): Kiolvassa a toplistát a getToplista() segítségével, majd hozzáfűzi a paraméterként kapott Eredményt. Visszaadja a kiegészített listát.

setToplista(): Soronként beírja a fájlba az Eredmények listáját, amit paraméterként megkapott.

textfieldBeallitas(): Kiírja a TextField-re az Eredményeket, soronként.

refresh(): Frissíti a toplistát, amennyiben új eredmény keletkezik. Az eredményt a megadott paraméterek segítségével hozza létre.

refresh(): (Paraméterek nélkül) A főmenü indulásakor frissíti a toplistát.

## 8. Az Eredmény osztályban

compare(): Összehasonlít két Eredményt, a kisebb idejű eredmény lesz a nagyobb.

## Felhasználói kézikönyv

A program indulásakor a főmenü fogadja a játékost. A név mezőbe tudja beírni a nevét, amivel esetleg a toplistán kíván szerepelni. A toplista mezőn láthatja az eddig elért legjobb eredményeket, a játékosok neveivel együtt. A főmenün szereplő JComboBox segítségével választhat egy nehézségi szintet, ami befolyásolni fogja a pálya méretét, és a benne található bombák számát. A játék gomb segítségével indíthatja el a kívánt beállításokkal az aknakeresőt.

A játék gomb megnyomása után elindul az aknakereső. A játékos láthatja a kezdés óta eltelt időt, a mezőket, illetve három fő gombot. Ezek az újra, a kattintás, és a zászlók. Alapvetően a játékos, ha rákattint egy mezőre, akkor a program felfedi azt a mezőt. Ha a játékos zászlókat kíván elhelyezni a pályán, akkor a zászlók gomb megnyomása után ezt megteheti. Ha ismét felfedni szeretné a mezőket, akkor a kattintás gombot kell megnyomnia. Az újra gomb megnyomásával új játék kezdődik.

## Use-case-ek

Cím	Főmenü használata
Leírás	A játékos vagy az egér bal gombjával kattint, illetve mozgatja az egeret
Aktorok	Játékos
Forgatókönyv	1. A játékos a név mezőre kattint, és beírja a nevét 2. A játékos kiválasztja a nehézségi fokozatot 3. A játékos a játék mezőre kattint
Alternatív forgatókönyv	1.A.1 A játékos nem ad meg nevet, ebben az esetben Névtelen lesz a név.

Cím	A tábla megfigyelése
Leírás	A játékos a képernyő megfigyelésével szerzi meg a megfelelő információkat a játékhoz
Aktorok	Játékos, Számítógép

<b>Forгатóкőnyv</b>	1. A számítógép megjeleníti az NxM-es mezőt 2. A játékos megfigyeli a mezőt.
---------------------	---

## Bemenetek, kimenetek leírása

A program a src mappában található Tolista.txt fájlból olvassa ki, illetve írja be az adatokat. Az egyes eredmények sorokkal vannak elválasztva, az egyes adattagok pedig szóközzel. Az első adattag a név, a második pedig az elért idő. (Ha a név mezőben szóközzel elválasztott szöveg van, akkor elromlik.) Például egy sor a txt fájlban: „Máté 50”. A txt fájl a src mappában van, a forráskódok mellett.