

# Házi feladat specifikáció

## Tartalomjegyzék

<b>Tartalomjegyzék.....</b>	<b>1</b>
<b>1: A feladat ismertetése, az Aknakereső játék bemutatása.....</b>	<b>2</b>
<b>2: Implementálandó funkciók, interakciók és állapotok .....</b>	<b>3</b>
<b>3: A megoldáshoz felhasználandó technikák illetve módszerek .....</b>	<b>4</b>

# Házi feladat specifikáció

## 1: A feladat ismertetése, az Aknakereső játék bemutatása

Az általam választott feladat egy “aknakereső” játék megvalósítása. Ez gyakorlatilag egy logikai játék, viszont előfordulhat olyan helyzet, ahol a játékosnak tippelnie kell. A lényege az, hogy egy aknamezőn kell megkeresni a véletlenszerűen elhelyezett robbanószerkezeteket. A játéknak 3 különböző, kezdéskor választható nehézségi szintje van. Az aknamező egy  $N \times N$  rublikával rendelkező táblaként van mevalósítva. Ezekre a rublikákra a továbbiakban a “mező” névvel fogok hivatkozni. A mezők a játék kezdetekor lefedett, ismeretlen állapotban vannak a játékos számára. A mezők felderítése az egér használtával, a mezőkre való klikkeléssel történik. Új játék indításánál az első klikk után a program a véletlenszerűen elhelyez a mezőkön a nehézségi szinttől függő fix számú aknát. Az első választott mező sosem lehet bomba. Amennyiben a játékos olyan mezőre kattint, amire a játék aknát helyezett el, a játék véget ér. Amennyiben nem, a mező attól függően, hogy hány bomba van az azt körülvevő mezőkön (8 mező, oldalra és átlóban) a következőket fedheti fel: üres mező, ami körül nincs bomba, vagy egy szám, ami a körülvevő bombák számát jelzi. A játékos jobb egér klikkel megjelölheti azokat a mezőket, ahol szerinte bomba van. Ha a játékos felfedte az összes olyan mezőt amin nem volt bomba, a játéknak vége, nyert. A grafikus felületen látszódik továbbá a játék kezdete óta eltelt idő másodpercben, illetve az elhelyezett bombák számának és a játékos által megjelölt mezők számának a különbsége. A program menüjéből elérhető az eddigi játékok eredménye, amely tartalmazza a játékos által megadott nevet, a nehézségi szintet, és a játék hosszát másodpercben. A nehézségi szintek:

- Könnyű (6x6 tábla, 10 akna)
- Közepes (8x8 tábla, 18 akna)
- Nehéz (10x10 tábla, 25 akna)

# Házi feladat specifikáció

## 2: Implementálandó funkciók, interakciók és állapotok

A program négy lehetséges állapot egyikében van, valamint a zárójelben jelzett állapotokba léphet belőle:

**-S1 Új játék (->S2,S4)**

**-S2 Játék folyamatban (->S1,S3)**

**-S3 Játék vége (->S1,S4)**

**-S4 Eredménytábla (->S1)**

A program induláskor új játék (S1) állapotban van. A játékos itt egy mezőben megadhatja a nevét, egy legördülő menüből nehézséget választhat, egy gombbal elindíthatja a játékot (->S2), valamint a menüsorból kiválasztva az ennek megfelelő pontot megjelenítheti az eredménytáblát (->S4).

A játékot elindítva (S2) a program a fenti leírásnak megfelelően játszható. A menüsorból a játék befejezhető (->S1), így új játék indítható. Ha bombára kattint, vagy felfedi az összes mezőt a játék véget ér (->S3).

Itt (S3), ha nyert, a program megjeleníti az elért eredményt, illetve eredményfájl keres, ha talál, kiegészíti azt az aktuális játék eredménnyel, ellenkező esetben új fájl hoz létre. Sikertelen játék esetén az eredmény nem kerül mentésre. A menüsorból új játék indítható (->S1), illetve megtekinthető az eredménytábla (->S4).

Az eredménytábla állapotban (S4) a program eredményfájlt keres, ha talál megjeleníti annak tartalmát, ha nem talál erről informálja a játékost. Az eredménytábla a fenti leírásnak megfelelő adatokat tartalmazza. A menüsorból új játék indítható (->S1).

A felhasználó bármelyik állapotban bezárhatja a programot.

# Házi feladat specifikáció

## 3: A megoldáshoz felhasználandó technikák illetve módszerek

Az eddig leírtakat Java nyelven, Swing grafikus felhasználói felület felhasználásával tervezem implementálni. A Swing eszközkészlete segítségével felépítek egy olyan felületet ami megfelel a fent leírtaknak. A programnak lesz egy állandó menüsora, a többi grafikus komponens az adott állapottól függően jelenik majd meg. A tábla egy kétdimenziós tömbben lesz eltárolva, a játék elején a bombák helyét véletlenszerűen generálja, a többi mező értékét ennek megfelelően számítja ki. Az egyes mezőkön az állapotukat reprezentáló PNG képeket jelenít meg. A megvalósítás során az egyes objektumok részére külön osztályt hozok létre. Tervezek létrehozni egy EREDMENY osztályt is, ami az elért eredményeket tárolja, és a Serializable interface segítségével képes fájlba kiírni illetve onnan visszaolvasni azokat.



Microsoft Minesweeper

Ezen a képen a játék egyik legelterjedtebb verziója látható, egy sikeres játék után. Az általam tervezett program nem fog megegyezni ezzel sem a felületében, sem a funkcionalitásában, a fent leírtakhoz szeretném tartani magamat.