

## Aknakereső

### 1 A FELADAT LEÍRÁSA

Feladatomban egy klasszikus aknakereső játék megvalósítását választottam. A játék célja a játékmezőn lévő összes akna elkerülése/megtalálása.

A játékmezőt egy kétdimenziós téglalap alakú négyzetrács alkotja. Minden négyzetrács (továbbiakban: cella) vagy aknát rejt, vagy pedig semmit. Azon cellák, amelyek nem tartalmaznak aknát, azt jelzik, hogy közvetlen szomszédságukban (fent, lent, jobbra, balra és átlósan) összesen hány akna (0-8 db) található. Az, hogy egy adott cella mit rejt, természetesen a játék kezdetekor nem látszik.

A játékos a játék kezdetén választhat, hogy mekkora legyen a játékmező (kicsi, közepes, nagy), ezáltal eldönti, hogy milyen nehéz legyen a játék (keves bomba, sok bomba).

A játékos egérrel kattinthat az egyes cellákra. Aknára való kattintás a játék elvesztését eredményezi. Ha a játékos egy olyan cellára kattint, ami nem akna, akkor a cellán megjelenik, hogy összesen hány aknával szomszédos. A legelső kattintott cella biztosan nem rejt aknát. A játékos a szerinte aknát tartalmazó cellákat zászlóval láthatja el, ekkor a cella kattintásra nem fog reagálni.

A játék akkor végződik győzelemmel, ha a játékos megtalálta az összes nem-akna cellát. Amennyiben a játékos gyorsan tisztította le a pályát, akkor felkerül a toplistára.

### 2 USE-CASE-EK

#### 2.1 USE-CASE DIAGRAM



#### 2.2 USE-CASE LEÍRÁSOK

Cím	Mező felfedése
Leírás	A játékos a játékmező egy cellájára kattint
Aktorok	Player
Főforgatókönyv	<ol style="list-style-type: none"> <li>1. A játékos egy nem-akna cellára kattint.</li> <li>2. Megjelenik a cellán, a szomszédos akna szám</li> </ol>
Alternatív forgatókönyv	<ol style="list-style-type: none"> <li>1.A.1. A játékos akna cellára kattint.</li> <li>1.A.2. Az akna felrobban, és a játéknak vége.</li> <li>1.A.3. Felfedésre kerül az egész játékmező.</li> </ol>

<b>Alternatív forgatókönyv</b>	<b>2.A.1.</b> A játékos az utolsó eltakart nem-akna cellára kattintott, és így a játékot megnyerte. <b>2.A.2.</b> Felfedésre kerül a játémező.
--------------------------------	---

<b>Cím</b>	<b>Mező megjelölése</b>
<b>Leírás</b>	A játékos a játémező egy celláját zászlóval jelöli meg.
<b>Aktorok</b>	Player
<b>Főforgatókönyv</b>	1. A játékos egy cellára zászlót helyez.

## 4 STRUKTURÁLIS LEÍRÁS

### 4.1 AZ OSZTÁLYOK LEÍRÁSA

Az osztályok létrehozása során igyekeztem a játék logikáját figyelembe venni. A tervezés során top-down megközelítést alkalmaztam, végiggondoltam, hogy melyik osztályra milyen felelősségeket szeretnék bízni, majd a szükséges főbb függvények megvalósításához konkretizáltam a szükséges további függvényeket.

#### 4.1.1 Cell

##### Felelősségek

Tárolja a játémező egyes celláinak paramétereit, tehát azt, hogy a játémezőn hol helyezkedik el, aknát rejt-e a cella, illetve körülötte hány cella tartalmaz aknát. Az osztály konstruktorában a sor, oszlop, és a játéktábla megadására van szükség.

##### Attribútumok

-mine: boolean	Igaz az értéke, ha a cella akna. Értéke kezdetben false.
-revealed: boolean	Igaz az értéke, ha a cellát a játékos már felfedte. Értéke kezdetben false.
-flagged: boolean	Igaz az értéke, ha a cellára a játékos zászlót helyezett. Értéke kezdetben false.
-col: int	Tárolja, hogy a cella a játémező melyik oszlopában szerepel, 0-tól indexelve.
-row: int	Tárolja, hogy a cella a játémező melyik sorában szerepel, 0-tól indexelve.
-mineNum: int	Tárolja, hogy hány szomszédos cellán van akna. Értéke -1, ha maga a cella tartalmaz aknát.
-table: Table	Az aktuális játémezőt tárolja.

**Metódusok**

+getMine(): boolean	Igaz értékkel tér vissza, ha a cellán akna van.
+getFlag(): boolean	Igaz értékkel tér vissza, ha a cellán zászló van.
+getMineNum(): int	Visszatér a cellával szomszédos cellákon található akna számával.
+getRevealed(): boolean	Igaz értékkel tér vissza, ha a cellát már felfedték.
+setMine(b: boolean)	A mine attribútum értékét a paraméterben átvett értékre állítja.
+flag()	Megváltoztatja a cella flagged attribútumának a logikai értékét.
+map()	Ha a cella nem akna, akkor beállítja a mineNum attribútum értékét aszerint, hogy a szomszédos cellák közül hány tartalmaz aknát. Amennyiben a cella maga is akna, akkor a mineNum értékét -1-re állítja.
+reveal(): boolean	Ha a cella még nincs felfedve, illetve ha nincs a cellán zászló, akkor felfedi a cellát: átállítja a revealed attribútum értékét true-ra, és amennyiben a cella közelében nincs akna, akkor a szomszédos mezőire is meghívja a reveal függvényt. Igaz értékkel tér vissza, ha éppen felfedtük a cellát, és aknát tartalmaz, minden más esetben visszatérési értéke hamis.

**4.1.2 Table****Felelősségek**

A játékmezőt jelentő osztály, létrehozza és tárolja a játékmező celláit. Biztosítja a cellák elérését a Game osztály számára. Ellenőrzi, hogy a játékot megnyerték-e. Az osztály konstruktorában a sorok, oszlopok és a táblán elhelyezkedő akna számát szükséges megadni. A konstruktor tölti fel a táblázatot cellákkal, azonban az akna elhelyezése később történik.

**Attribútumok**

-col: int	Tárolja, hogy hány oszlopa van a játékmezőnek.
-row: int	Tárolja, hogy hány sora van a játékmezőnek.
-mineNum: int	Tárolja, hogy a játékmező létrehozásakor összesen mennyi aknát tartalmaz.
-matrix[][]: Cell	Kétdimenziós Cell típusú tömbben tárolja a játékmezőn lévő cellákat.
-rand: Random	Véletlenszerű érték generálásáért felelős attribútum.

**Metódusok**

+fillTable(starter: Cell)	Véletlenszerűen elhelyez mineNum számú aknát a játékmezőn. Figyel arra, hogy a paraméterként átvett kezdő mezőn ne szerepeljen akna.
+getCell(r: int, c: int): Cell	Visszatér a játékmező r. sorában és c. oszlopában lévő cellával. Amennyiben a paraméterként kapott indexek érvénytelenek, null értékkel tér vissza.
+revealAll()	Felfedi a játékmező összes celláját. Amennyiben a cellán zászló lett elhelyezve, úgy ezt eltávolítja.
+reveal(r: int, c: int): boolean	Felfedi az indexként átvett cellát. True értékkel tér vissza, ha éppen felfedtük a cellát, és az aknát rejtett, egyéb esetben visszatérési értéke false.
+flag(r: int, c: int)	Zászlót helyez el vagy vesz fel paraméterként kapott celláról.
+write()	Kiírja a játékmezőt konzolra.
+checkEnd(): boolean	Igaz értékkel tér vissza, ha a játékot megnyerték, vagyis minden nem-akna cellát felfedtek.

**4.1.3 Game****Felelősségek**

A játék elindításaért, menetéért és kimeneteléért felelős osztály. Ezek mellett itt tárolódik az aktuális játékos elért eredménye.

Az osztály konstruktorában kerül inicializálásra az időzítő, a grafikus felület, illetve a ranglisták is. A konstruktornak nincsenek paraméterei.

**Attribútumok**

-cols: int	Tárolja, hogy a játékmezőnek hány oszlopa van.
-mines: int	Tárolja, hogy a játékmezőn hány akna található.
-row: int	Tárolja, hogy a játékmezőnek hány sora van.
-difficulty: String	Stringként tárolja, hogy milyen nehézségű a játék (easy, medium, hard).
-name: String	Tárolja az aktuális játékos nevét.
-end: boolean	Igaz az értéke, ha a játéknak valamilyen módon vége. Értéke kezdetben false.
-timer: Timer	A játékidő méréséért felelős attribútum.
-time: int	A játékidő hossza másodpercben kifejezve.
-status: boolean	Igaz az értéke, ha a játéknak vége van, és megnyerték. Értéke kezdetben false.
-table: Table	Az aktuális játékmezőt tárolja.

-frame: GUI	A játék grafikus megjelenítéséért felelős osztályt tárolja.
-scoreboards: HashMap<String, Scoreboard>	A játékokon elért legjobb eredményeket tárolja nehézségtől függően.
-result: Score	Az adott játék adatait (a jelenlegi játékos nevét és a játékidőt) tárolja.

### Metódusok

+main(args[]: String)	A program belépési pontja.
+startGame(diff: String)	Létrehozza a játékmezőt a paraméterben átvett nehézségtől függően, és elindítja az időzítőt. A paraméter három értéket vehet fel: easy, medium és hard.
-endGame()	A játék befejezéséért felelős függvény. Felfedi az összes cellát a játékmezőn, leállítja az időzítőt, és létrehozza a result attribútumot a játékos nevével és játékidőjével. Beállítja az end attribútum értékét true-ra.
-winGame()	A játék megnyeréséért felelős függvény. A játék lezárásához meghívja az endGame metódust, és beállítja a status attribútum értékét true-ra. Hozzáadja a játékos eredményét a megfelelő nehézségű ranglistához.
-loseGame()	A játék elvesztéséért felelős függvény. Beállítja a status attribútum értékét false-ra.
+getRow(): int	Visszatér a játékmező sorainak számával.
+getCol(): int	Visszatér a játékmező oszlopainak számával.
+reveal(r: int, c: int)	Felfordítja a paraméterként átvett indexű cellát. Amennyiben a cella aknát rejtett, akkor meghívja a loseGame függvényt. Ha a cella nem akna volt, és ez volt az utolsó, eddig felfordítatlan nem-akna cella, meghívja a winGame függvényt.
+flag(r: int, c: int)	Zászlót helyez el vagy vesz el a paraméterként kapott indexű celláról.
+getTable(): Table	Visszatér a játékmezővel.
+setName(n: String)	Beállítja a játékos nevét.
+getEnd(): boolean	Visszatér az end attribútum értékével.
+gameWon(): boolean	Igaz értékkel tér vissza, ha a játéknak vége, és azt megnyerték, egyébként visszatérési értéke false.
+getTime(): String	Visszaadja a játékidőt String típusként.

+getScoreboard(diff: String): Scoreboard	Visszatér a paraméterként átvett nehézségű eredmény ranglistával. A paraméter három értéket vehet fel: easy, medium és hard.
--	--

#### 4.1.4 Score

##### Felelősség

Legfőképp adattárolásra használt osztály, amely eltárolja egy játékos nevét, és az általa elért játékidőt. Impementálja a Serializable interfészt.

Az osztály konstruktorában szükséges a játékos nevét, a játékidőt, illetve a játék nehézségét megadni.

##### Attribútumok

-serialVersionUID: long	Az osztály serializációjának verziókezeléséért felel.
-name: String	Eltárolja a játékos nevét.
-time: int	Eltárolja a játékos által elért játékidőt másodpercben.
-difficulty: String	Eltárolja, hogy milyen nehézségi fokozaton került elérésre az eredmény. A változó három értéket vehet fel: easy, medium és hard.

##### Metódusok

+getTime(): int	Visszaadja a játékidőt másodpercben.
+convertTime(): String	Visszatér egy Stringgel, amely formázott, perc:másodperc alakban tartalmazza a játékidőt.
+toString(): String	Visszatér egy Stringgel, amely az elért eredményt formázotán tárolja.

#### 4.1.5 Scoreboard

##### Felelősségek

A legjobb 10 eredmény nyilvántartásáért felelős osztály. Impementálja a Serializable interfészt.

Az osztály konstruktorában annak a fájlnak a nevét szükséges megadni, ahol a a ranglista tárolása megvalósul.

##### Attribútumok

-serialVersionUID: long	Az osztály serializációjának verziókezeléséért felel.
-------------------------	---

-file: String	Tartalmazza a korábbi eredményeket tartalmazó fájl nevét. Ugyanebbe a fájlba kerül kiírásra a frissült ranglista. A fájlnev a konstruktorban kerül átvételre.
-list: ArrayList<Score>	Tartalmazza a legjobb 10 eredményt.

### Metódusok

+add(newScore: Score)	Megvizsgálja, hogy a paraméterként kapott eredmény felfér-e a ranglistára. Amennyiben igen, a lista legrosszabb eredményét törli, és helyére az új eredményt helyezi. Ezek után legjobb időeredménytől legrosszabb időeredményig rendezi a listát, majd meghívja a saveData függvényt.
-addData()	Feltölti a listát egy fájlból, melynek nevét a file attribútum tárolja.
-saveData()	Elmenti a file attribútumban tárolt nevű fájlba a listát.
+toString(): String	Visszatér egy String típussal, amely tartalmazza a rangistán szerelő eredményeket rendezve.
+getPlace(place: int)	Visszatér egy olyan Stringgel, amely a place-dik helyezett nevével és játékidőjével.

### 4.1.6 ScoreComparator

#### Felelősségek

Két Score típusú változót az elért játékidő alapján összehasonlít.

#### Attribútumok

-

#### Metódusok

+compare(s1: Score, s2: Score): int	Összehasonlít két paraméterül kapott értéket. Negatív értékkel tér vissza, ha az első paraméter a kisebb, 0-val ha a két paraméter megegyezik, és pozitív értékkel, ha az első paraméter nagyobb.
-------------------------------------	---

### 4.1.7 GUI

#### Felelősségek

A játék megjelenítéséért és a felhasználóval történő kommunikációért felelős osztály, a JFrame osztály leszármazottja.

Az osztály konstruktorában szükséges megadni az aktuális játékot.

#### Attribútumok

-nameField: JTextField	A játékos nevét bekérő szövegdoz.
-startButton: JButton	A játék elindításához szükséges gomb.

-difficultyBox: JComboBox	A játék nehézségét meghatározó választódoboz. Három érték közül lehet választani: easy, medium és hard.
-layout: GridLayout	A grafikus megjelenítés elrendezését határozza meg.
-difficulty: String	A kiválasztott nehézsége a játéknak.
-edition: JCheckBox	Annak a meghatározására szolgáló checkbox, hogy a játékos normál vagy covid edition játékot szeretne játszani.
-result: JDialog	A játék végén megjelenő párbeszédablak.
-init: JPanel	A játék kezdetekor megjelenő panel, amely a játék elindításához szükséges elemeket tartalmazza: nehézség beállítása, név megadása, edition kiválasztása, és az indítógomb.
-mineField: JPanel	A játékmezőt tartalmazó panel.
-endPanel: JPanel	A játék végén megjelenő párbeszédablak tartalma. Tartalmazza a megfelelő nehézségű ranglistát, a játékos által elért eredményt, és az új játék gombot.
-menuBar: JMenuBar	A menüsört tartalmazó elem.
-menuDialog: JDialog	A menün kiválasztott opció megjelenítéséért felelős párbeszédablak.
-close: JButton	A menuDialog párbeszédablakokat bezáró gomb.
-firstClick: boolean	Igaz az értéke, ha a következő kattintás a játék első kattintása lesz. Kezdeti értéke igaz.
-covidEdition: boolean	Igaz az értéke, ha az edition checkbox a játék indításakor be volt pipálva.
-buttonSize: int	A gombok mérete pixelben.
-BOMB: String	A normál edition során megjelenő bomba karakterért felelős String.
-FLAG: String	A normál edition során megjelenő zászló karakterért felelős String.
-bombCovid: Icon	A covid edition során megjelenő vírus iconért felelős Icon.
-flagCovid: Icon	A covid edition során megjelenő maszk iconért felelős Icon.
-empty: Icon	Üres icon.
-buttonColors[]: Color	A játék során a cellák számozásához felhasznált színeket tartalmazó, Color típusú tömb.
-game: Game	Eltárolja az aktuális játékot.



**Metódusok:**

-setMenu()	Beállítja, hogy mi jelenjen meg az ablakok menüsorában.
-setMineField()	A játéklemez megfelelő megjelenítésért felelős függvény.
-setEndPanel()	A játék végén megjelenő párbeszédablak feltöltéséért felelős függvény.
-drawGrid()	A játéklemez kirajzolásáért felelős függvény.
-openRules()	A szabályok megnyitásáért felelős függvény.
-openScoreboard(diff: String)	A paraméterként átvett nehézségű ranglista megjelenítéséért felelős függvény. A paraméter három értéket vehet fel: easy, medium és hard.

**4.1.8 MenuActionListener****Felelősség**

A kiválasztott menüpont megjelenítéséért felelős figyelő. Implementálja az ActionListener interfészt.

**Attribútumok**

-

**Metódusok**

actionPerformed(ae: ActionEvent)	Ez a függvény hívódik meg, ha a felhasználó valamilyen menüpontra kattint. A választott menüpont alapján meghívja a megfelelő párbeszédablak megjelenítéséért felelős függvényt.
----------------------------------	--

**4.1.9 ButtonActionListener****Felelősségek**

A megnyomott gombok megfelelő viselkedéséért felelős figyelő. Megvalósítja az ActionListener interfészt.

**Attribútumok**

-

**Metódusok**

actionPerformed(ae: ActionListener)	Ez a függvény hívódik meg, ha a felhasználó egy gombra kattint a grafikus felületen. Meghívja a gomb lenyomása miatt elvárt ablak megjelenítéséért felelős függvényt.
-------------------------------------	---

#### 4.1.10 CellClickMouseListener

##### Felelősségek

A felhasználó játékmезőn történő kattintásai regisztrálásáért felelős figyelő. Ez az osztály biztosítja, hogy a grafikus felület a játék logikájától elvárt módon működjön. Megvalósítja a MouseListener interfészt. Az interfész függvényei közül a mouseClicked függvény került implementálásra.

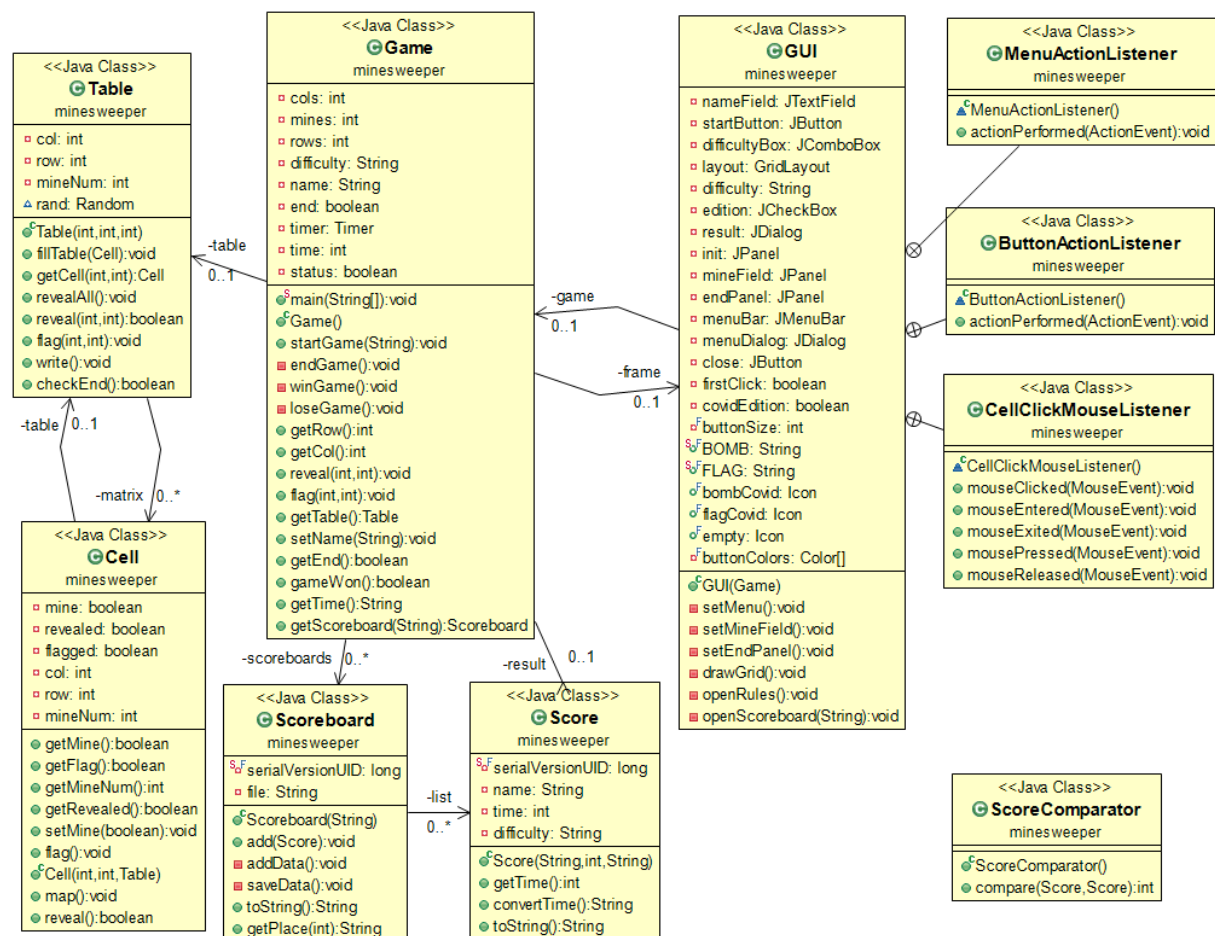
##### Attribútumok

-

##### Metódusok

+mouseClicked(e: MouseEvent)	Kapcsolatot létesít a grafikus felület és a játék megfelelő cellája között. Felfedetlen cellákon bal egérgomb hatására felfedi a cellát amennyiben nincs rajta zászló, jobb egérgomb hatására pedig elhelyez egy zászlót a cellán, vagy felveszi a korábban lerakott zászlót.
------------------------------	---

## 4.2 OSZTÁLYDIAGRAM



---

## 5 FÁJLOK SZERKEZETE

---

A programnak a megfelelő működéshez képfájllra van szüksége. Ezek a minesweeper nevű mappában találhatóak, mask\_covid.png, illetve mine\_covid.png néven, az scr mappa mellett

A játékok során elért eredmények ranglistái szövegfájlokban kerülnek eltárolásra. Minden ranglistához külön szövegfájl tartozik: easyScoreboard.txt a könnyű nehézséghez, mediumScoreboard.txt a közepes nehézséghez, illetve hardScoreboard.txt a legnehezebb nehézséghez. Ezeket a fájlokat a képek mellett a minesweeper mappában kell tárolni. Amennyiben valamely szövegfájl nem található ezen a helyen, úgy a játék automatikusan létrehoz egyet, ha eredmény kerül fel az adott nehézségű ranglistára.

---

## 6 USER MANUAL

---

A játéktér egy kétdimenziós téglalap alakú négyzetrács alkotja. Minden négyzetrács (továbbiakban: cella) vagy aknát rejt, vagy pedig semmit. Azon cellák, amelyek nem tartalmaznak aknát, azt jelzik, hogy közvetlen szomszédságukban (fent, lent, jobbra, balra és átlósan) összesen hány akna (0-8 db) található. Az, hogy egy adott cella mit rejt, természetesen a játék kezdetekor nem látszik.

A játékos a játék kezdetén választhat, hogy mekkora legyen a játéktér (kicsi, közepes, nagy), ezáltal eldönti, hogy milyen nehéz legyen a játék (keves bomba, sok bomba). A választást egy lenyíló menü segítségével teheti meg.

A játékos egérrel kattinthat az egyes cellákra. Aknára való kattintás a játék elvesztését eredményezi. Ha a játékos egy olyan cellára kattint, ami nem akna, akkor a cellán megjelenik, hogy összesen hány aknával szomszédos. A legelső kattintott cella biztosan nem rejt aknát. A játékos a szerinte aknát tartalmazó cellákat zászlóval láthatja el, ekkor a cella kattintásra nem fog reagálni.

A játék akkor végződik győzelemmel, ha a játékos megtalálta az összes nem-akna cellát. Amennyiben a játékos gyorsan tisztította le a pályát, akkor felkerül a toplistára. A játéklistán való megjelenítéshez a játékos megadhatja a játék indításakor a nevét a megfelelő szövegdobozban.

Kétféle játékmód között lehet választani: hagyományos aknakereső aknákkal és zászlókkal, illetve 2020-as koronavírus mód között, amiben aknák helyett vírusok, zászlók helyett pedig maszkok szereplenek. A játékos a játék indítása előtt választhat játékmódot egy választódoboz bepipálásával.

Új játékosoknak lehetősége nyílik a szabályok megismerésére a menüsor Help gombját, azon belül pedig a Rules opciót kiválasztva. A játékosok megtekinthetik a ranglistákat a Scorebord gombot kiválasztva a menüsoron, ezek után pedig választhat, hogy melyik nehézségű ranglistát szeretné megtekinteni.