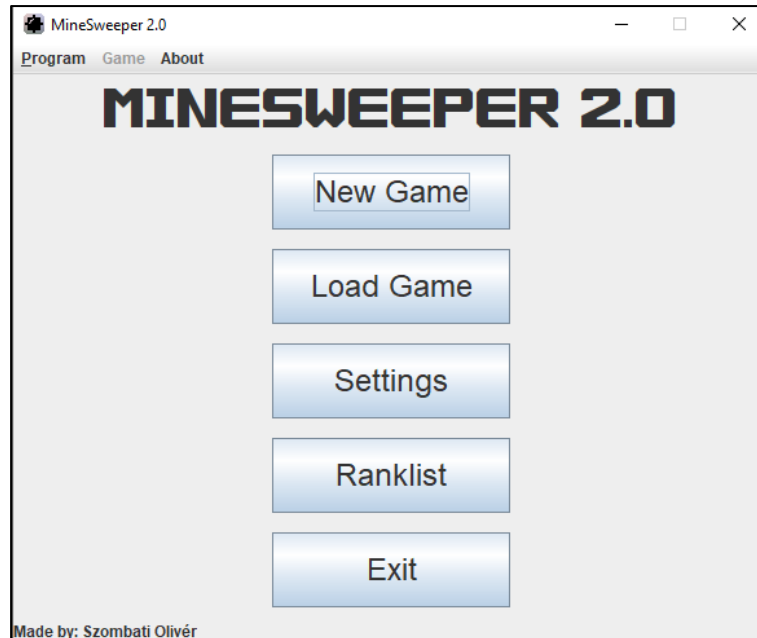


MineSweeper 2.0 felhasználói kézikönyv

1. Program indítása

A program indítását követően a következő ablak fogad minket:

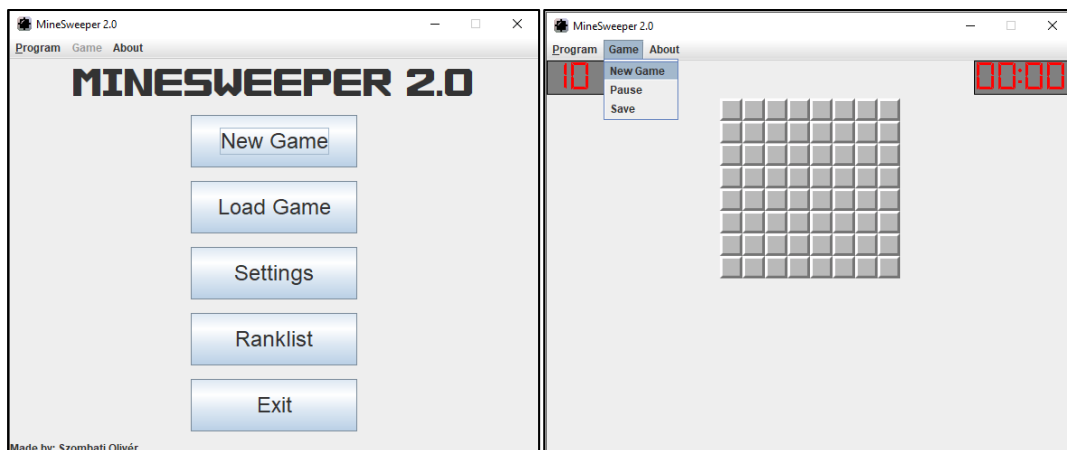


Megjegyzés: első indításkor felugorhat egy hibaüzenet, ami szerint a rekordokat nem sikerült betölteni. Ebben az esetben nincs semmi teendő, következő indításnál nem fog megjelenni.

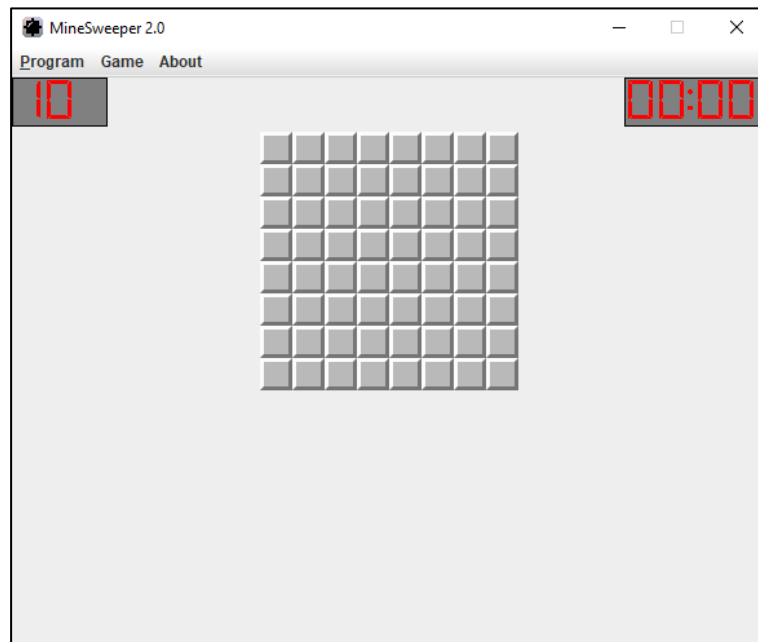
Megjegyzés: ha nem sikerült betölteni a játékhoz szükséges erőforrásokat, akkor hibaüzenetet kapunk.

2. Játék kezelése

2.1. Új játékot a "New Game" címkéjű gombbal lehet indítani, vagy a menü "Game/New Game" opciójával. (Az utóbbi csak játék közben érhető el)

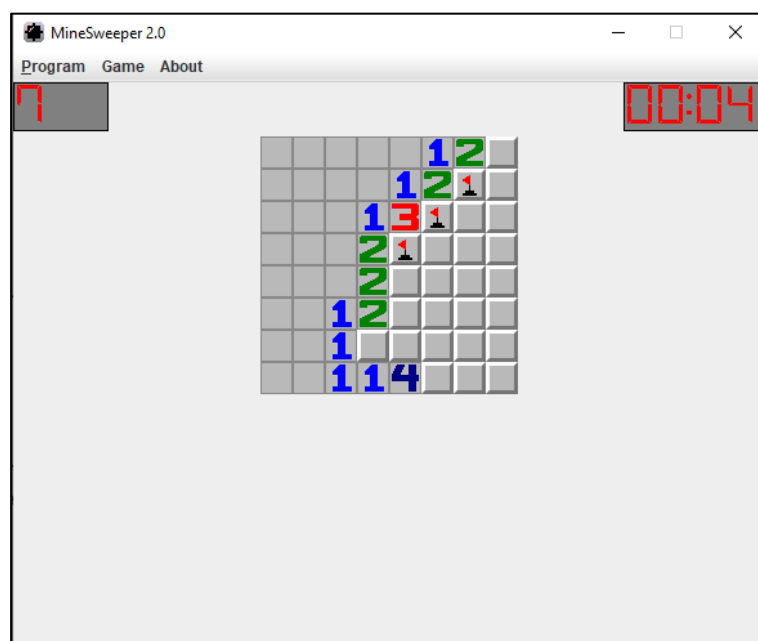


2.2. Ha új játékot indítottunk, ez a kép fogad minket:





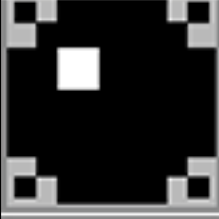



Miután elindítottuk a játékot (ehhez a játékmezőn belül kell kattintani) elindul a stopper, ami mérni fogja az időnket.

2.4. Mezőt **felfedni** a bal egérgombbal, **megjelölni** zászlóval a jobb egérgombbal lehet. Megjelölés után védve van a menő, nem lehet felfedni. A jelölést leszedni a jobb egérgomb újbóli megnyomásával lehet.



2.5. A pályán a következő bombák lehetnek:

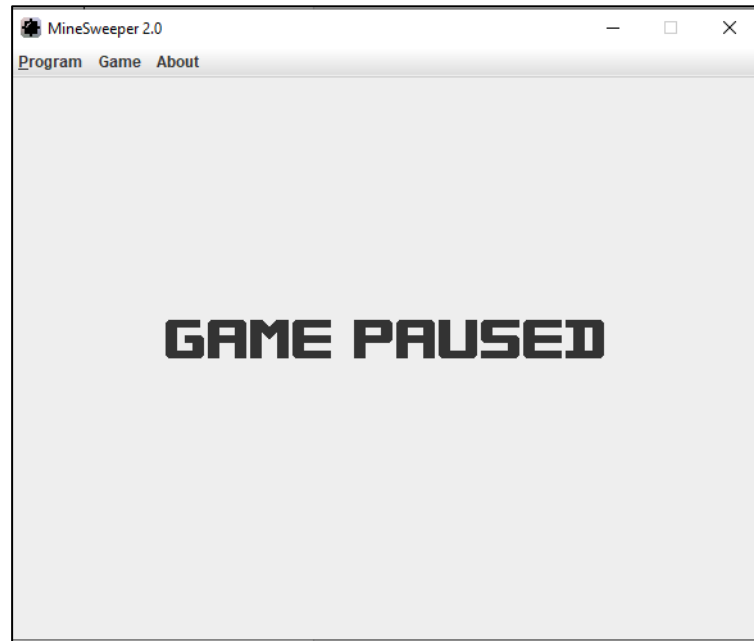
	CLUSTER BOMB Ha van még elegendő hely a pályán, akkor elhelyez 1-3 új bombát. Azonban ha nincs már elég hely, felrobban.
	DIFUSED BOMB Hatástalanított bomba. Nem csinál semmit, viszont ugyanúgy bombának számít.
	RESET BOMB Újragenerálja a pályát, új bombákat helyez el, viszont az időzítő ugyanúgy megy tovább.
	RESET FLAG BOMB Törli a már lerakott zászlókat a pályáról.
	BIG BOMB Nagy bomba, azaz ez kettőt ér. Amezők 2 bombaként érzékelik.
	BOMB A megszokott bombák. Felfedés elején véget ér a játék. A játék végét okozó bomba piros háttérű lesz.

2.6. A nem bomba mezők azt tárolják, hogy hány bomba van körülöt-tük.



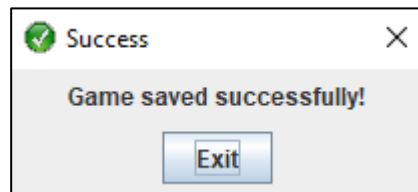
Megjegyzés: alapvetően csak 8 bomba lehetne egy cella körül, viszont ebben a változatban lehetséges, hogy csupa Big Bomb van egy cella körül. Szóval ha egy cella körül több, mint 8 bomba van, akkor a cellát felfedve a fenti "?" ikon jelenik meg.

2.7. Ha szüneteltetni akarjuk a játékot, akkor a fejléc "Game/Pause" opcióját kell választani. Ekkor az időzítő is megáll, illetve erre vált át a nézet:



2.8. Folytatáshoz ugyanezt a lehetőséget kell választani, de most a címke "Continue" feliratú lesz.

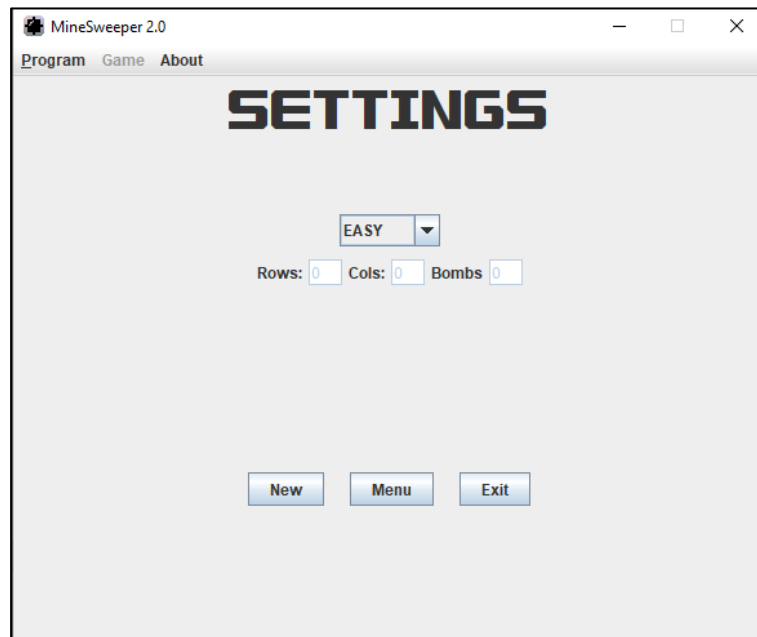
2.9. Játékállást menteni a "Game/Save" gombbal lehet. Ekkor megáll a játék, és sikeres mentés esetén ezt az üzenetet kapjuk:



2.10. A fejléc "About/Help" opcióját választva információkat kaphatunk az eredeti játék működéséről.

3. Nehézség állítása

3.1. A menürendszer "Settings" gombjára nyomva vagy a fejléc "Program/Settings" opcióját választva áttérünk a beállításokba:



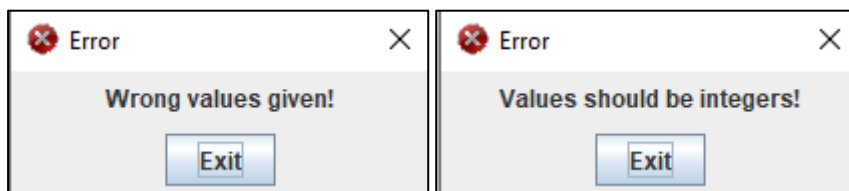
3.2. 3 alapnehézség van:

- Könnyű - 8x8-as tábla 10 bombával
- Normál - 16x16-os tábla 40 bombával
- Nehéz - 16x32-es tábla 99 bombával

Ezekén kívül megadható saját nehézség saját bombaszámmal, az értékek beállításakor a következőkre kell figyelni:

- A sorok és oszlopok száma külön-külön legalább 8 és legfeljebb 32 kell legyen
- A bombák száma meg legalább a cellák számának 10%-a kell legyen.

Hibás érték (nem egész vagy nem felel meg a fentieknek) esetén a következő üzeneteket kapjuk:

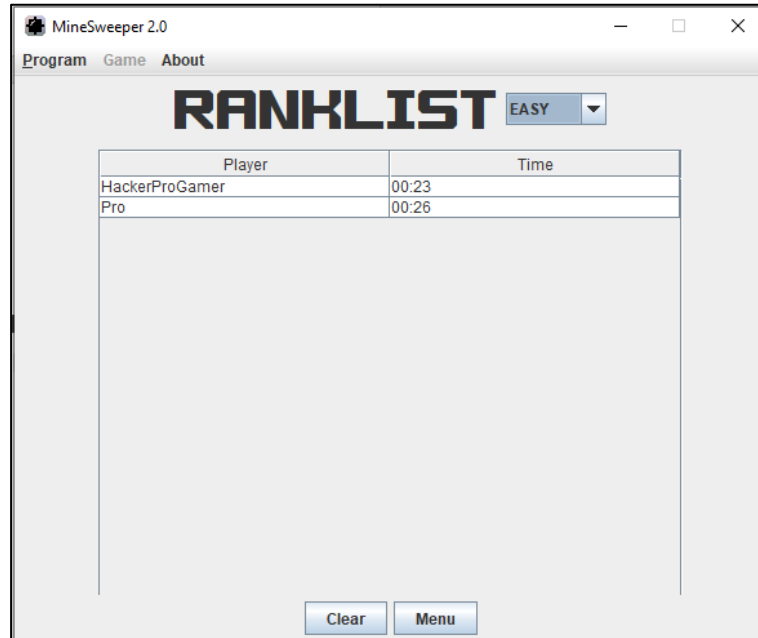


3.3 A beállításokból a menübe vissza lehet térni, illetve új játékot is lehet indítani.

4. Ranglista megtekintése

4.1. A ranglisták megtekintéséhez a menürendszer "Ranklist" gombjára vagy a fejléc "Program/Ranklist" opciójára kell nyomni.

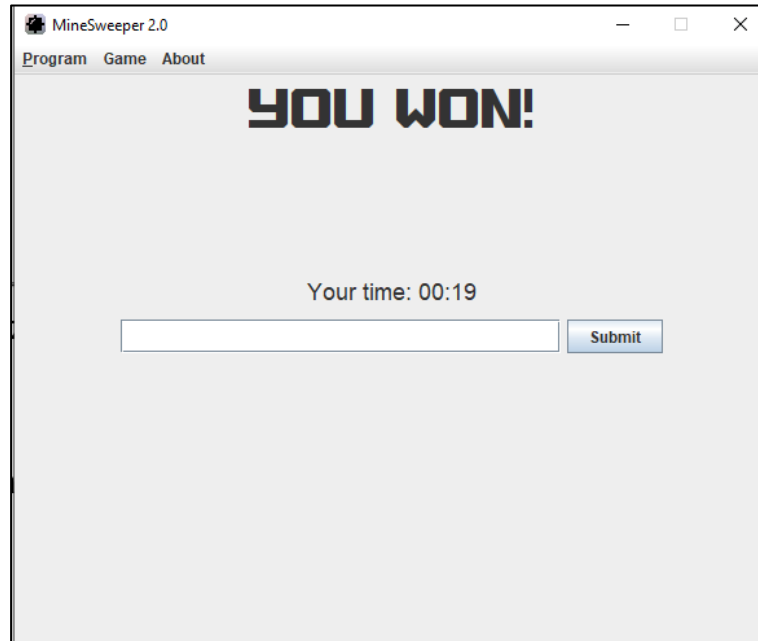
A ranglista nézet így néz ki alpból:



4.2. A legördülő listából kiválasztható, hogy melyik nehézség rangsorát jelenítsük meg.

5. Rekord felvétele

5.1. Ha a játékos megnyeri a játékot, ezt az ablakot fogja látni:



5.2. A nevét beírva (ami minimum 1 maximum 33 karakteres lehet) és a "Submit" gombot megnyomva elmentheti az idejét.

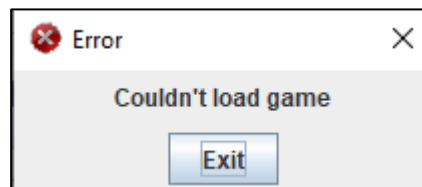
Megjegyzés: saját nezéség idejét nem lehet elmenteni.

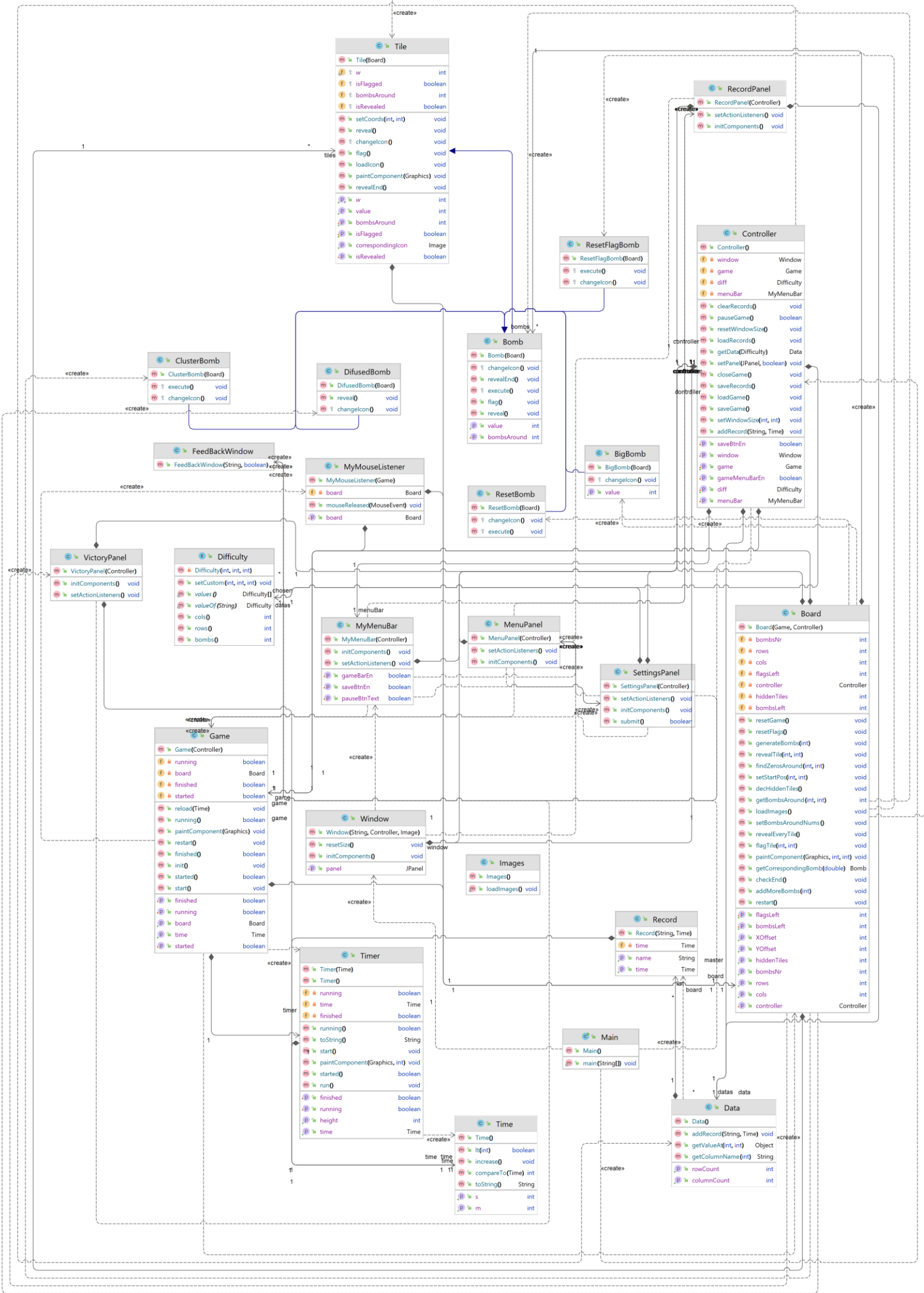
6. Játék betöltése

6.1. Ha van korábbi mentés, a menürendszer "Load Game" gombjával betölthető.

Ekkor visszatér a mentett játék állapotába, megállítva.

6.2. Ha nincs korábbi mentése vagy sérült a mentés, az alábbi hiba-üzenetet kapja:





MineSweeper 2.0

Dokumentáció
Készítette Doxygen 1.9.1

1. Hierarchikus mutató	1
1.1. Osztályhierarchia	1
2. Osztálymutató	3
2.1. Osztálylista	3
3. Osztályok dokumentációja	5
3.1. backend.BigBomb osztályreferencia	5
3.1.1. Részletes leírás	6
3.1.2. Konstruktork és destruktorok dokumentációja	6
3.1.2.1. BigBomb()	6
3.1.3. Tagfüggvények dokumentációja	7
3.1.3.1. changelcon()	7
3.1.3.2. getValue()	7
3.2. backend.Board osztályreferencia	8
3.2.1. Részletes leírás	9
3.2.2. Konstruktork és destruktorok dokumentációja	9
3.2.2.1. Board()	9
3.2.3. Tagfüggvények dokumentációja	10
3.2.3.1. addMoreBombs()	10
3.2.3.2. checkEnd()	10
3.2.3.3. decHiddenTiles()	10
3.2.3.4. findZerosAround()	10
3.2.3.5. flagTile()	11
3.2.3.6. generateBombs()	11
3.2.3.7. getBombsAround()	11
3.2.3.8. getBombsLeft()	12
3.2.3.9. getBombsNr()	12
3.2.3.10. getCols()	12
3.2.3.11. getCorrespondingBomb()	12
3.2.3.12. getFlagsLeft()	12
3.2.3.13. getHiddenTiles()	12
3.2.3.14. getRows()	13
3.2.3.15. getXOffset()	13
3.2.3.16. getYOffset()	13
3.2.3.17. loadImages()	13
3.2.3.18. paintComponent()	13
3.2.3.19. resetFlags()	13
3.2.3.20. resetGame()	14
3.2.3.21. restart()	14
3.2.3.22. revealEveryTile()	14
3.2.3.23. revealTile()	14
3.2.3.24. setBombsAroundNums()	14

3.2.3.25.	setBombsLeft()	14
3.2.3.26.	setController()	15
3.2.3.27.	setFlagsLeft()	15
3.2.3.28.	setStartPos()	15
3.3.	backend.Bomb osztályreferencia	15
3.3.1.	Részletes leírás	16
3.3.2.	Konstruktorok és destruktork dokumentációja	16
3.3.2.1.	Bomb()	16
3.3.3.	Tagfüggvények dokumentációja	17
3.3.3.1.	changelcon()	17
3.3.3.2.	execute()	17
3.3.3.3.	flag()	17
3.3.3.4.	getValue()	18
3.3.3.5.	reveal()	18
3.3.3.6.	revealEnd()	18
3.3.3.7.	setBombsAround()	18
3.4.	backend.ClusterBomb osztályreferencia	19
3.4.1.	Részletes leírás	20
3.4.2.	Konstruktorok és destruktork dokumentációja	20
3.4.2.1.	ClusterBomb()	20
3.4.3.	Tagfüggvények dokumentációja	20
3.4.3.1.	changelcon()	20
3.4.3.2.	execute()	21
3.5.	backend.Controller osztályreferencia	21
3.5.1.	Részletes leírás	21
3.5.2.	Konstruktorok és destruktork dokumentációja	21
3.5.2.1.	Controller()	22
3.5.3.	Tagfüggvények dokumentációja	22
3.5.3.1.	addRecord()	22
3.5.3.2.	clearRecords()	22
3.5.3.3.	closeGame()	22
3.5.3.4.	getData()	22
3.5.3.5.	getDiff()	23
3.5.3.6.	getGame()	23
3.5.3.7.	loadGame()	23
3.5.3.8.	loadRecords()	23
3.5.3.9.	pauseGame()	23
3.5.3.10.	resetWindowSize()	24
3.5.3.11.	saveGame()	24
3.5.3.12.	saveRecords()	24
3.5.3.13.	setDiff()	24
3.5.3.14.	setGame()	24

3.5.3.15. setGameMenuBarEn()	24
3.5.3.16. setMenuBar()	24
3.5.3.17. setPanel()	24
3.5.3.18. setSaveBtnEn()	25
3.5.3.19. setWindow()	25
3.5.3.20. setWindowSize()	25
3.6. backend.Data osztályreferencia	26
3.6.1. Részletes leírás	26
3.6.2. Tagfüggvények dokumentációja	26
3.6.2.1. addRecord()	27
3.6.2.2. getColumnCount()	27
3.6.2.3. getColumnName()	27
3.6.2.4. getRowCount()	27
3.6.2.5. getValueAt()	27
3.7. backend.Difficulty felsoroló referencia	28
3.7.1. Részletes leírás	28
3.7.2. Tagfüggvények dokumentációja	28
3.7.2.1. bombs()	28
3.7.2.2. cols()	29
3.7.2.3. rows()	29
3.7.2.4. setCustom()	29
3.7.3. Adattagok dokumentációja	29
3.7.3.1. EASY	29
3.8. backend.DifusedBomb osztályreferencia	30
3.8.1. Részletes leírás	31
3.8.2. Konstruktorok és destruktorkok dokumentációja	31
3.8.2.1. DifusedBomb()	31
3.8.3. Tagfüggvények dokumentációja	31
3.8.3.1. changelcon()	31
3.8.3.2. reveal()	32
3.9. frontend.FeedBackWindow osztályreferencia	32
3.9.1. Részletes leírás	33
3.9.2. Konstruktorok és destruktorkok dokumentációja	33
3.9.2.1. FeedBackWindow()	33
3.10. frontend.Game osztályreferencia	33
3.10.1. Részletes leírás	34
3.10.2. Konstruktorok és destruktorkok dokumentációja	34
3.10.2.1. Game()	34
3.10.3. Tagfüggvények dokumentációja	35
3.10.3.1. finished()	35
3.10.3.2. getBoard()	35
3.10.3.3. getTime()	35

3.10.3.4. init()	35
3.10.3.5. paintComponent()	35
3.10.3.6. reload()	36
3.10.3.7. restart()	36
3.10.3.8. running()	36
3.10.3.9. setFinished()	36
3.10.3.10. setRunning()	36
3.10.3.11. setStarted()	37
3.10.3.12. start()	37
3.10.3.13. started()	37
3.11. backend.Images osztályreferencia	38
3.11.1. Részletes leírás	39
3.11.2. Tagfüggvények dokumentációja	39
3.11.2.1. loadImages()	39
3.11.3. Adattagok dokumentációja	39
3.11.3.1. bigBomb	39
3.11.3.2. bomb	39
3.11.3.3. clusterBomb	39
3.11.3.4. difusedBomb	39
3.11.3.5. error	40
3.11.3.6. flag	40
3.11.3.7. flag2	40
3.11.3.8. hTile	40
3.11.3.9. icon	40
3.11.3.10. MineSweeperFont	40
3.11.3.11. numbers	40
3.11.3.12. redBomb	40
3.11.3.13. resetBomb	41
3.11.3.14. resetFlagBomb	41
3.11.3.15. success	41
3.11.3.16. timerFont	41
3.11.3.17. unknown	41
3.12. backend.Main osztályreferencia	41
3.12.1. Részletes leírás	41
3.12.2. Tagfüggvények dokumentációja	42
3.12.2.1. main()	42
3.13. frontend.MenuPanel osztályreferencia	42
3.13.1. Részletes leírás	43
3.13.2. Konstruktorok és destruktorok dokumentációja	43
3.13.2.1. MenuPanel()	43
3.13.3. Tagfüggvények dokumentációja	43
3.13.3.1. initComponents()	43

3.13.3.2. <code>setActionListeners()</code>	43
3.14. frontend.MyMenuBar osztályreferencia	44
3.14.1. Részletes leírás	44
3.14.2. Konstruktorkok és destruktorkok dokumentációja	45
3.14.2.1. <code>MyMenuBar()</code>	45
3.14.3. Tagfüggvények dokumentációja	45
3.14.3.1. <code>initComponents()</code>	45
3.14.3.2. <code>setActionListeners()</code>	45
3.14.3.3. <code>setGameBarEn()</code>	45
3.14.3.4. <code>setPauseBtnText()</code>	46
3.14.3.5. <code>setSaveBtnEn()</code>	46
3.15. frontend.MyMouseListener osztályreferencia	46
3.15.1. Részletes leírás	47
3.15.2. Konstruktorkok és destruktorkok dokumentációja	47
3.15.2.1. <code>MouseListener()</code>	47
3.15.3. Tagfüggvények dokumentációja	47
3.15.3.1. <code>mouseReleased()</code>	47
3.15.3.2. <code>setBoard()</code>	48
3.16. backend.Record osztályreferencia	49
3.16.1. Részletes leírás	50
3.16.2. Konstruktorkok és destruktorkok dokumentációja	50
3.16.2.1. <code>Record()</code>	50
3.16.3. Tagfüggvények dokumentációja	50
3.16.3.1. <code>getName()</code>	50
3.16.3.2. <code>getTime()</code>	50
3.17. frontend.RecordPanel osztályreferencia	51
3.17.1. Részletes leírás	51
3.17.2. Konstruktorkok és destruktorkok dokumentációja	52
3.17.2.1. <code>RecordPanel()</code>	52
3.17.3. Tagfüggvények dokumentációja	52
3.17.3.1. <code>initComponents()</code>	52
3.17.3.2. <code>setActionListeners()</code>	52
3.18. backend.ResetBomb osztályreferencia	53
3.18.1. Részletes leírás	54
3.18.2. Konstruktorkok és destruktorkok dokumentációja	54
3.18.2.1. <code>ResetBomb()</code>	54
3.18.3. Tagfüggvények dokumentációja	54
3.18.3.1. <code>changelcon()</code>	54
3.18.3.2. <code>execute()</code>	54
3.19. backend.ResetFlagBomb osztályreferencia	55
3.19.1. Részletes leírás	56
3.19.2. Konstruktorkok és destruktorkok dokumentációja	56

3.19.2.1. ResetFlagBomb()	56
3.19.3. Tagfüggvények dokumentációja	56
3.19.3.1. changelcon()	56
3.19.3.2. execute()	56
3.20. frontend.SettingsPanel osztályreferencia	57
3.20.1. Részletes leírás	57
3.20.2. Konstruktorok és destruktork dokumentációja	57
3.20.2.1. SettingsPanel()	57
3.20.3. Tagfüggvények dokumentációja	58
3.20.3.1. initComponents()	58
3.20.3.2. setActionListeners()	58
3.20.3.3. submit()	58
3.21. backend.Tile osztályreferencia	59
3.21.1. Részletes leírás	60
3.21.2. Konstruktorok és destruktork dokumentációja	60
3.21.2.1. Tile()	60
3.21.3. Tagfüggvények dokumentációja	61
3.21.3.1. changelcon()	61
3.21.3.2. flag()	61
3.21.3.3. getBombsAround()	61
3.21.3.4. getCorrespondingIcon()	61
3.21.3.5. getValue()	62
3.21.3.6. getW()	62
3.21.3.7. isFlagged()	62
3.21.3.8. loadIcon()	62
3.21.3.9. paintComponent()	62
3.21.3.10. reveal()	63
3.21.3.11. revealEnd()	63
3.21.3.12. setBombsAround()	63
3.21.3.13. setCoords()	63
3.21.3.14. setFlagged()	64
3.21.3.15. setRevealed()	64
3.21.4. Adattagok dokumentációja	64
3.21.4.1. bombsAround	64
3.21.4.2. icon	64
3.21.4.3. isFlagged	64
3.21.4.4. isRevealed	64
3.21.4.5. master	65
3.21.4.6. w	65
3.21.4.7. x	65
3.22. backend.Time osztályreferencia	65
3.22.1. Részletes leírás	66

3.22.2. Konstruktorkok és destruktorkok dokumentációja	66
3.22.2.1. Time()	66
3.22.3. Tagfüggvények dokumentációja	66
3.22.3.1. compareTo()	66
3.22.3.2. getM()	67
3.22.3.3. getS()	67
3.22.3.4. increase()	67
3.22.3.5. lt()	67
3.22.3.6. toString()	68
3.23. frontend.Timer osztályreferencia	68
3.23.1. Részletes leírás	69
3.23.2. Konstruktorkok és destruktorkok dokumentációja	69
3.23.2.1. Timer() [1/2]	69
3.23.2.2. Timer() [2/2]	69
3.23.3. Tagfüggvények dokumentációja	69
3.23.3.1. getHeight()	70
3.23.3.2. getTime()	70
3.23.3.3. paintComponent()	70
3.23.3.4. run()	70
3.23.3.5. running()	70
3.23.3.6. setFinished()	71
3.23.3.7. setRunning()	71
3.23.3.8. start()	71
3.23.3.9. started()	71
3.23.3.10. toString()	71
3.24. frontend.VictoryPanel osztályreferencia	72
3.24.1. Részletes leírás	72
3.24.2. Konstruktorkok és destruktorkok dokumentációja	72
3.24.2.1. VictoryPanel()	73
3.24.3. Tagfüggvények dokumentációja	73
3.24.3.1. initComponents()	73
3.24.3.2. setActionListeners()	73
3.25. frontend.Window osztályreferencia	73
3.25.1. Részletes leírás	74
3.25.2. Konstruktorkok és destruktorkok dokumentációja	74
3.25.2.1. Window()	74
3.25.3. Tagfüggvények dokumentációja	75
3.25.3.1. initComponents()	75
3.25.3.2. resetSize()	75
3.25.3.3. setPanel()	75

1. fejezet

Hierarchikus mutató

1.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

backend.Controller	21
backend.Difficulty	28
backend.Images	38
backend.Main	41
Thread	
frontend.Timer	68
AbstractTableModel	
backend.Data	26
JDialog	
frontend.FeedBackWindow	32
JFrame	
frontend.Window	73
JMenuBar	
frontend.MyMenuBar	44
JPanel	
frontend.Game	33
frontend.MenuPanel	42
frontend.RecordPanel	51
frontend.SettingsPanel	57
frontend.VictoryPanel	72
MouseAdapter	
frontend.MyMouseListener	46
Serializable	
backend.Board	8
backend.Record	49
backend.Tile	59
backend.Bomb	15
backend.BigBomb	5
backend.ClusterBomb	19
backend.DifusedBomb	30
backend.ResetBomb	53
backend.ResetFlagBomb	55
backend.Time	65
frontend.Game	33
frontend.MyMouseListener	46
frontend.Timer	68

2. fejezet

Osztálymutató

2.1. Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

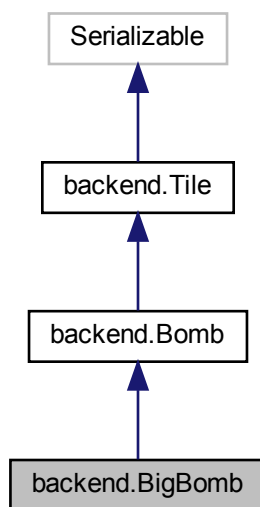
backend.BigBomb	5
backend.Board	8
backend.Bomb	15
backend.ClusterBomb	19
backend.Controller	21
backend.Data	26
backend.Difficulty	28
backend.DifusedBomb	30
frontend.FeedBackWindow	32
frontend.Game	33
backend.Images	38
backend.Main	41
frontend.MenuPanel	42
frontend.MyMenuBar	44
frontend.MyMouseListener	46
backend.Record	49
frontend.RecordPanel	51
backend.ResetBomb	53
backend.ResetFlagBomb	55
frontend.SettingsPanel	57
backend.Tile	59
backend.Time	65
frontend.Timer	68
frontend.VictoryPanel	72
frontend.Window	73

3. fejezet

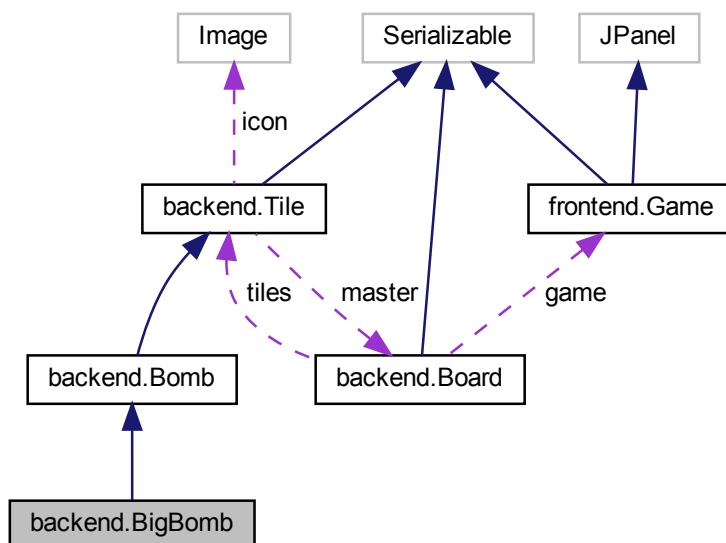
Osztályok dokumentációja

3.1. backend.BigBomb osztályreferencia

A backend.BigBomb osztály származási diagramja:



A backend.BigBomb osztály együttműködési diagramja:



Publikus tagfüggvények

- `BigBomb (Board b)`
- `int getValue ()`

Védett tagfüggvények

- `void changelcon ()`

További örökölt tagok

3.1.1. Részletes leírás

Nagy bomba osztály. Felrobban, ha rányomnak, de a sima bombával ellentétben 2-nek számít.

3.1.2. Konstruktorkok és destruktorkok dokumentációja

3.1.2.1. BigBomb()

```
backend.BigBomb.BigBomb (
    Board b )
```

A létrehozása azonos a sima bombáéval.

Paraméterek

<i>b,az</i>	öt tartalmazó tábla.
-------------	----------------------

3.1.3. Tagfüggvények dokumentációja

3.1.3.1. changelcon()

```
void backend.BigBomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: [backend.Bomb](#).

3.1.3.2. getValue()

```
int backend.BigBomb.getValue ( )
```

A getBombsAround függvénynél használatos.

Visszatérési érték

2 bombának megfelelő az értéke.

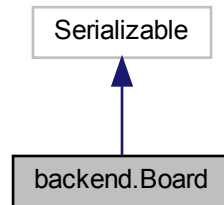
Újrimplementált ősök: [backend.Bomb](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

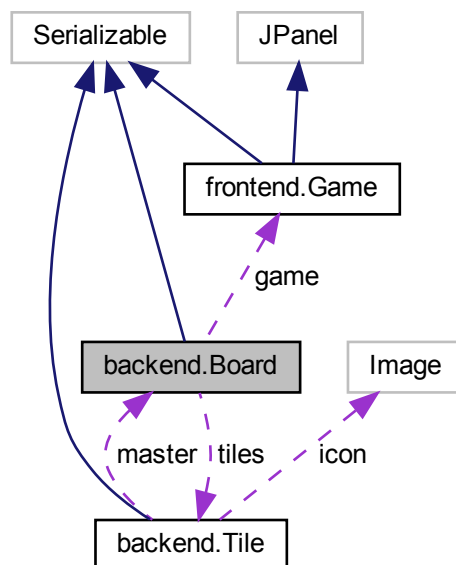
- src/backend/BigBomb.java

3.2. backend.Board osztályreferencia

A backend.Board osztály származási diagramja:



A backend.Board osztály együttműködési diagramja:



Publikus tagfüggvények

- `Board` (`Game` g, `Controller` controller)
- `int getYOffset ()`
- `int getXOffset ()`
- `int getRows ()`
- `int getCols ()`

- int [getBombsNr](#) ()
- int [getBombsLeft](#) ()
- int [getFlagsLeft](#) ()
- int [getHiddenTiles](#) ()
- void [setStartPos](#) (int x, int y)
- void [setFlagsLeft](#) (int i)
- void [setBombsLeft](#) (int i)
- void [setController](#) ([Controller](#) c)
- void [decHiddenTiles](#) ()
- void [resetGame](#) ()
- void [restart](#) ()
- void [generateBombs](#) (int startBombNr)
- [Bomb](#) [getCorrespondingBomb](#) (double chance)
- void [setBombsAroundNums](#) ()
- int [getBombsAround](#) (int row, int col)
- void [revealEveryTile](#) ()
- void [findZerosAround](#) (int row, int col)
- void [loadImages](#) ()
- void [resetFlags](#) ()
- void [revealTile](#) (int row, int col)
- void [flagTile](#) (int col, int row)
- void [checkEnd](#) ()
- void [addMoreBombs](#) (int db)
- void [paintComponent](#) (Graphics g, int startX, int startY)

3.2.1. Részletes leírás

A cellákat tartalmazó tábla osztálya.

3.2.2. Konstruktorkok és destruktorok dokumentációja

3.2.2.1. Board()

```
backend.Board.Board (
    Game g,
    Controller controller )
```

A tábla konstruktora. Beállítja a megfelelő értékeket és generál egy (még bombák nélküli) táblát.

Paraméterek

<i>g</i>	Az őt tartalmazó játék.
<i>controller</i>	Az őt (is) irányító controller.

3.2.3. Tagfüggvények dokumentációja

3.2.3.1. addMoreBombs()

```
void backend.Board.addMoreBombs (
    int db )
```

Extra bombák elhelyezése a pályán. Növeli az elhelyezhető zászlók és a hátralévő bombák számát. Az össz bombaszámot átállítja. Az ikonokat újratölti és a sima cellák körüli bombák számát újraszámolja.

Paraméterek

<i>db</i>	Ennyi db új bombát kell elhelyezni.
-----------	-------------------------------------

3.2.3.2. checkEnd()

```
void backend.Board.checkEnd ( )
```

Ellenőrzi, hogy nyert-e a játékos. Ez akkor van, ha már nincs jelöletlen (fedett) bomba vagy ha már csak annyi fedett cella van, ahány bomba maradt.

3.2.3.3. decHiddenTiles()

```
void backend.Board.decHiddenTiles ( )
```

Eggyel csökkenti a még fedett cellák számát.

3.2.3.4. findZerosAround()

```
void backend.Board.findZerosAround (
    int row,
    int col )
```

Flood-fill algoritmus cellák körüli üres vagy 1 értékű cellák felfedésére.

Paraméterek

<i>row</i>	A cella sorszáma
<i>col</i>	A cella oszlopszáma

3.2.3.5. flagTile()

```
void backend.Board.flagTile (
    int col,
    int row )
```

Megjelöl zászlóval egy mezőt. Itt is ellenőrzi, hogy nyert-e a játékos.

Paraméterek

<i>col</i>	A megjelölésre szánt cella oszlopszáma.
<i>row</i>	A megjelölésre szánt cella sorszáma.

3.2.3.6. generateBombs()

```
void backend.Board.generateBombs (
    int startBombNr )
```

Bombák generálása. Annyiszor fut le, ahány bombát el kell helyezni. Kiválaszt egy véletlenszerű sort és oszlopot, és ha az a cella nincs felfedve, nem bomba (még) és nem a kezdőkattintás helyén van, akkor egy véletlenszerű bombává teszi azt.

Paraméterek

<i>startBombNr</i>	A kezdeti bombaszám. Játék kezdetekor 0, pálya újratekésésénél a régi bombaszám értéke.
--------------------	---

3.2.3.7. getBombsAround()

```
int backend.Board.getBombsAround (
    int row,
    int col )
```

Egy cella körüli bombák számát számolja meg. Végigmegy a cella körül és összeszámolja a bombákat.

Paraméterek

<i>row</i>	A cella sorszáma
<i>col</i>	A cella oszlopszáma

Visszatérési érték

A megadott cella körüli bombák száma

3.2.3.8. getBombsLeft()

```
int backend.Board.getBombsLeft ( )
```

Visszaadja a még hátralévő bombák számát.

3.2.3.9. getBombsNr()

```
int backend.Board.getBombsNr ( )
```

Visszaadja a tábla bombáinak számát.

3.2.3.10. getCols()

```
int backend.Board.getCols ( )
```

Visszaadja a tábla oszlopainak számát.

3.2.3.11. getCorrespondingBomb()

```
Bomb backend.Board.getCorrespondingBomb (
    double chance )
```

Egy véletlen szám alapján visszaad egy bombát. Hatástalanított bombára 5%, ClusterBomba 10%, zászló eltűntető bombára 10%, nagy bombára 20%, pálya újrakezdő bombára 20% és sima bombára 35% az esély.

Paraméterek

<i>chance</i>	A valószínűség.
---------------	-----------------

Visszatérési érték

A megfelelő bomba.

3.2.3.12. getFlagsLeft()

```
int backend.Board.getFlagsLeft ( )
```

Visszaadja a még elhelyezhető zászlók számát.

3.2.3.13. getHiddenTiles()

```
int backend.Board.getHiddenTiles ( )
```

Visszaadja a még fedett cellák számát.

3.2.3.14. getRows()

```
int backend.Board.getRows ( )
```

Visszaadja a tábla sorainak számát.

3.2.3.15. getXOffset()

```
int backend.Board.getXOffset ( )
```

Visszaadja a tábla függőleges eltolását.

3.2.3.16. getYOffset()

```
int backend.Board.getYOffset ( )
```

Visszaadja a tábla vertikális eltolását.

3.2.3.17. loadImages()

```
void backend.Board.loadImages ( )
```

A cellák ikonját tölti be. Játék betöltésekor illetve extra bombák hozzáadásakor használt.

3.2.3.18. paintComponent()

```
void backend.Board.paintComponent (
    Graphics g,
    int startX,
    int startY )
```

A tábla és celláinak kirajzolása.

Paraméterek

<i>g</i>	A grafikáért felelős paraméter.
<i>startX</i>	A tábla kezdeti x koordinátája. A Game adja át.
<i>startY</i>	A tábla kezdeti y koordinátája. A Game adja át.

3.2.3.19. resetFlags()

```
void backend.Board.resetFlags ( )
```

Leszedi az összes elhelyezett zászlót a pályáról.

3.2.3.20. resetGame()

```
void backend.Board.resetGame ( )
```

Visszaállítja a játék kezdeti állapotát (nem generálja újra)

3.2.3.21. restart()

```
void backend.Board.restart ( )
```

Újragenerálja a táblát. Beállítja a megfelelő értékeket majd csinál egy oszlop*sor nagyságú táblát csupa sima cellával.

3.2.3.22. revealEveryTile()

```
void backend.Board.revealEveryTile ( )
```

Minden cellát felfed. A játék elvesztésekor használt.

3.2.3.23. revealTile()

```
void backend.Board.revealTile (
    int row,
    int col )
```

Felfedi a megadott cellát. Ha bomba vagy 2-nél több értékű cella, akkor simán felfedi, egyéb esetben meghívja a findZerosAround függvényt. Bármely felfedés esetén ellenőrzi, hogy nyert-e a játékos.

Paraméterek

<i>row</i>	A felfedésre szánt cella sorszáma.
<i>col</i>	A felfedésre szánt cella oszlopszáma.

3.2.3.24. setBombsAroundNums()

```
void backend.Board.setBombsAroundNums ( )
```

A cellák körüli bombaszámok beállítása. Végigmegy az összes cellán és lekérdezi a körülöttük lévő bombák számát (getBombsAround), ezt állítja be.

3.2.3.25. setBombsLeft()

```
void backend.Board.setBombsLeft (
    int i )
```

Beállítja a még hátralévő bombák számát

3.2.3.26. setController()

```
void backend.Board.setController (
    Controller c )
```

Beállítja a táblát kezelő controllert

3.2.3.27. setFlagsLeft()

```
void backend.Board.setFlagsLeft (
    int i )
```

Beállítja a még elhelyezhető zászlók számát

3.2.3.28. setStartPos()

```
void backend.Board.setStartPos (
    int x,
    int y )
```

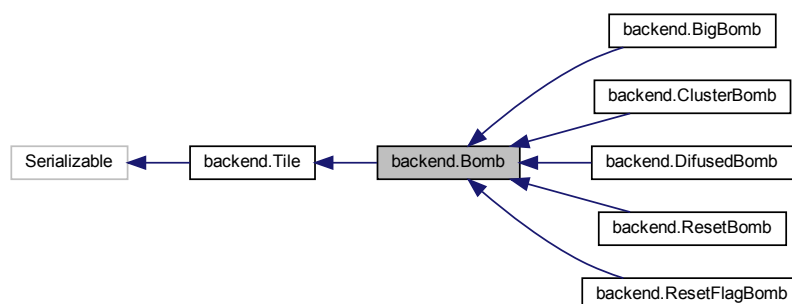
Beállítja az elsőként felfedett cella oszlopát és sorát

Ez a dokumentáció az osztályról a következő fájl alapján készült:

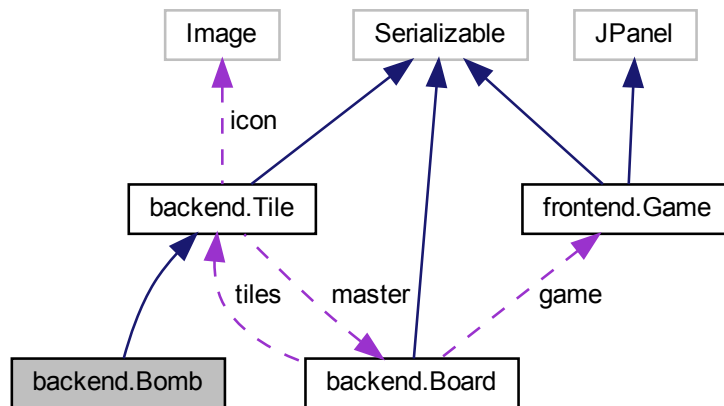
- src/backend/Board.java

3.3. backend.Bomb osztályreferencia

A backend.Bomb osztály származási diagramja:



A backend.Bomb osztály együttműködési diagramja:



Publikus tagfüggvények

- **Bomb** (**Board** b)
- void **reveal** ()
- void **revealEnd** ()
- void **flag** ()
- void **setBombsAround** (int b)
- int **getValue** ()

Védett tagfüggvények

- void **changeIcon** ()
- void **execute** ()

További örökölt tagok

3.3.1. Részletes leírás

Bomba ősosztály. Sima mezőből származik, viszont ezt ha felfedik, akkor felrobban a pálya és a játéknak vége.

3.3.2. Konstruktorkok és destruktorkok dokumentációja

3.3.2.1. Bomb()

```
backend.Bomb.Bomb (
    Board b )
```

A létrehozása azonos a sima mezőével.

Paraméterek

<i>b,az</i>	őt tartalmazó tábla.
-------------	----------------------

3.3.3. Tagfüggvények dokumentációja

3.3.3.1. changelcon()

```
void backend.Bomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: [backend.Tile](#).

Újrimplementáló leszármazottak: [backend.ResetFlagBomb](#), [backend.ResetBomb](#), [backend.DifusedBomb](#), [backend.ClusterBomb](#) és [backend.BigBomb](#).

3.3.3.2. execute()

```
void backend.Bomb.execute ( ) [protected]
```

Bomba aktiválása. Alapvetően felfedi az összes cellát és ezáltal véget ér a játék.

Újrimplementáló leszármazottak: [backend.ResetFlagBomb](#), [backend.ResetBomb](#) és [backend.ClusterBomb](#).

3.3.3.3. flag()

```
void backend.Bomb.flag ( )
```

Mező megjelölése. A hagyományos megjelöléssel ellentétben ha bombát jelölünk meg, akkor az a hátralévő bombák számát is csökkenti.

Újrimplementált ősök: [backend.Tile](#).

3.3.3.4. `getValue()`

```
int backend.Bomb.getValue ( )
```

A `getBombsAround` függvénynél használatos.

Visszatérési érték

1 bombának megfelelő az értéke.

Újrimplementált ősök: [backend.Tile](#).

Újrimplementáló leszármazottak: [backend.BigBomb](#).

3.3.3.5. `reveal()`

```
void backend.Bomb.reveal ( )
```

Sima felfedéskor beállítja, hogy ő az első bomba, ami a robbanást okozta (ezáltal piros háttérű lesz) és ha még nincs felfedve (ez a végtelen ciklus elkerüléséhez kell) akkor megjeleníti (`revealEnd`) és aktiválja (`execute`).

Újrimplementált ősök: [backend.Tile](#).

Újrimplementáló leszármazottak: [backend.DifusedBomb](#).

3.3.3.6. `revealEnd()`

```
void backend.Bomb.revealEnd ( )
```

Végső felfedés. Ez a felfedés általában a játék végét jelzi. Ha nincs megjelölve zászlóval, akkor megváltoztatja az ikonját, beállítja, hogy már fel van fedve és csökkenti a tábla fedett mezőinek számát.

Újrimplementált ősök: [backend.Tile](#).

3.3.3.7. `setBombsAround()`

```
void backend.Bomb.setBombsAround (
    int b )
```

A bombáknál a `setBombsAround` függvény minden esetben -1-et állít be, mivel bombáknál nem számít, hogy hány bomba van körülöttük.

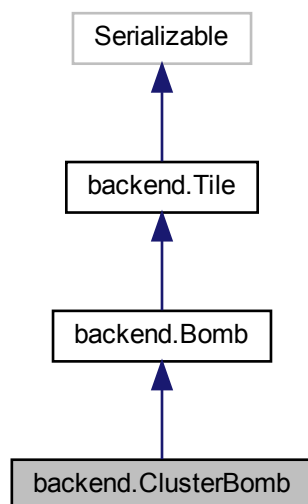
Újrimplementált ősök: [backend.Tile](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

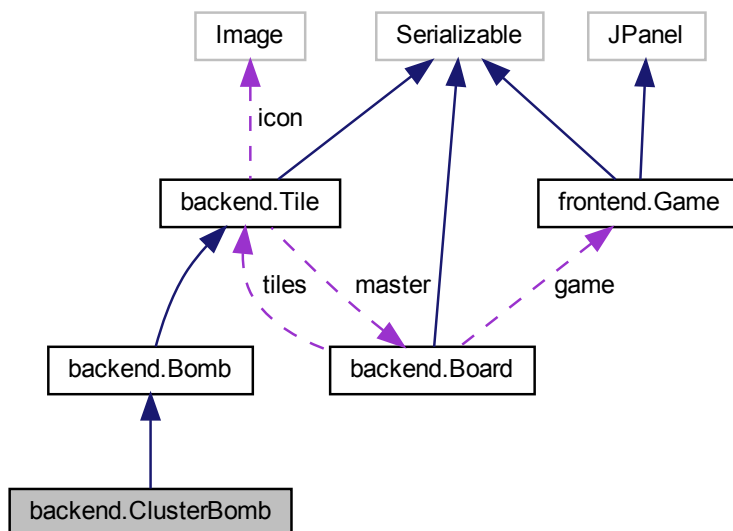
- `src/backend/Bomb.java`

3.4. backend.ClusterBomb osztályreferencia

A backend.ClusterBomb osztály származási diagramja:



A backend.ClusterBomb osztály együttműködési diagramja:



Publikus tagfüggvények

- `ClusterBomb` (`Board b`)

Védett tagfüggvények

- void [changeIcon](#) ()
- void [execute](#) ()

További örökölt tagok

3.4.1. Részletes leírás

Cluster bomba osztály. Ha van elég hely a táblán, akkor felfedéskor elhelyez még 1-3 új bombát. Egyéb esetben felrobban.

3.4.2. Konstruktorkok és destruktorok dokumentációja

3.4.2.1. ClusterBomb()

```
backend.ClusterBomb.ClusterBomb (
    Board b )
```

A létrehozása azonos a sima bombáéval.

Paraméterek

<i>b,az</i>	őt tartalmazó tábla.
-------------	----------------------

3.4.3. Tagfüggvények dokumentációja

3.4.3.1. changeIcon()

```
void backend.ClusterBomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: [backend.Bomb](#).

3.4.3.2. execute()

```
void backend.ClusterBomb.execute ( ) [protected]
```

Felfedéskor először meghatároz egy random számot 1 és 3 között - ez lesz az új bombák száma. Ezt követően megvizsgálja, hogy van-e elég hely még a táblán, ha nincs, akkor felfedi az összes mezőt. Egyéb esetben a táblának meghívja az extra bomba elhelyezési függvényét, illetve csökkenti a saját értékét a zászlók és hátralévő bombák tekintetében. 1 és 3 között ad hozzá új bombát

Újraimplementált ősök: [backend.Bomb](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- src/backend/ClusterBomb.java

3.5. backend.Controller osztályreferencia

Publikus tagfüggvények

- [Controller](#) ()
- void [setGame](#) ([Game](#) g)
- [Game](#) [getGame](#) ()
- void [setGameMenuBarEn](#) (boolean b)
- void [setSaveBtnEn](#) (boolean b)
- void [setMenuBar](#) ([MyMenuBar](#) m)
- void [setWindow](#) ([Window](#) w)
- [Difficulty](#) [getDiff](#) ()
- void [setDiff](#) ([Difficulty](#) d)
- void [setPanel](#) (JPanel panel, boolean needsResize)
- void [setWindowSize](#) (int w, int h)
- void [resetWindowSize](#) ()
- void [closeGame](#) ()
- void [addRecord](#) (String playerName, [Time](#) time)
- boolean [pauseGame](#) ()
- void [loadRecords](#) ()
- void [saveRecords](#) ()
- [Data](#) [getData](#) ([Difficulty](#) d)
- void [clearRecords](#) ()
- void [saveGame](#) ()
- void [loadGame](#) ()

3.5.1. Részletes leírás

[Controller](#) osztály, ami összeköttetést teremt a megjelenítés és a logika között. A legtöbb osztály vele van kapcsolatban, ő állítja az értékeiket.

3.5.2. Konstruktorkok és destruktorkok dokumentációja

3.5.2.1. Controller()

```
backend.Controller.Controller ( )
```

A controller konstruktora. Alapból könnyű nehézséget állítunk be rá, ezt a későbbiekben lehet módosítani. Ezt követően a rekordok HashMapjét állítja be alapértékre. (Ha esetleg nem sikerülne betölteni)

3.5.3. Tagfüggvények dokumentációja

3.5.3.1. addRecord()

```
void backend.Controller.addRecord (
    String playerName,
    Time time )
```

Rekord felvétele. A megfelelő nehézséghez adja hozzá.

Paraméterek

<i>playerName</i>	A játékos neve.
<i>time</i>	A játékos ideje.

3.5.3.2. clearRecords()

```
void backend.Controller.clearRecords ( )
```

Törli a rekordokat az adatbázisából.

3.5.3.3. closeGame()

```
void backend.Controller.closeGame ( )
```

Bezárja a játékot. A sima System.exit(0) nem jó mentésekhez

3.5.3.4. getData()

```
Data backend.Controller.getData (
    Difficulty d )
```

A megfelelő nehézség adatának lekérdezése.

Paraméterek

<i>d</i>	A kért nehézség.
----------	------------------

Visszatérési érték

A nehézség adata.

3.5.3.5. getDiff()

```
Difficulty backend.Controller.getDiff ( )
```

Visszaadja a controller által kezelt nehézséget. (A SettingsPanel és a VictoryPanelnél használt főként)

3.5.3.6. getGame()

```
Game backend.Controller.getGame ( )
```

Visszaadja a controller által kezelt játékot. (VictoryPanelnél használt.)

3.5.3.7. loadGame()

```
void backend.Controller.loadGame ( )
```

Betölti az elmentett játékot. Ha nincs korábbi mentés vagy sérült a fájl, akkor hibaüzenetet dob. Egyébként visszatölti a játékot a megállított állapotába.

3.5.3.8. loadRecords()

```
void backend.Controller.loadRecords ( )
```

Rekordok betöltése a program indításakor. Ha nem sikerül, hibaüzenetet ad.

3.5.3.9. pauseGame()

```
boolean backend.Controller.pauseGame ( )
```

Megállítja a játékot

Visszatérési érték

A játék most meg van-e állítva.

3.5.3.10. resetWindowSize()

```
void backend.Controller.resetWindowSize ( )
```

Visszaállítja az ablak méretét az alapértelmezett értékére

3.5.3.11. saveGame()

```
void backend.Controller.saveGame ( )
```

Elmenti a játékállást. Először megállítja a játék futását, aztán megpróbálja elmenteni. A sikertől függően visszaad egy feedback üzenetet.

3.5.3.12. saveRecords()

```
void backend.Controller.saveRecords ( )
```

Rekorok elmentése a program bezárásakor.

3.5.3.13. setDiff()

```
void backend.Controller.setDiff (
    Difficulty d )
```

Beállítja a controller által kezelt nehézséget

3.5.3.14. setGame()

```
void backend.Controller.setGame (
    Game g )
```

Beállítja a controller által kezelt játékot

3.5.3.15. setGameMenuBarEn()

```
void backend.Controller.setGameMenuBarEn (
    boolean b )
```

Beállítja a controller által kezelt fejléc játék mezőjének engedélyezettségét

3.5.3.16. setMenuBar()

```
void backend.Controller.setMenuBar (
    MyMenuBar m )
```

Beállítja a controller által kezelt fejlécet

3.5.3.17. setPanel()

```
void backend.Controller.setPanel (
    JPanel panel,
    boolean needsResize )
```

Beállítja az ablak Paneljét.

Paraméterek

<i>panel</i>	A beállítandó JPanel
<i>needsResize</i>	Azt jelöli, hogy szükséges lehet-e az ablak átméretezése. (Leginkább ha játékot kezdünk)

Egyszerűbb itt mindig engedélyezni, aztán máshol letiltani

3.5.3.18. setSaveBtnEn()

```
void backend.Controller.setSaveBtnEn (
    boolean b )
```

Beállítja a mentés gomb engedélyezettségét

3.5.3.19. setWindow()

```
void backend.Controller.setWindow (
    Window w )
```

Beállítja a controller által kezelt ablakot

3.5.3.20. setWindowSize()

```
void backend.Controller.setWindowSize (
    int w,
    int h )
```

Beállítja az ablak méretét.

Paraméterek

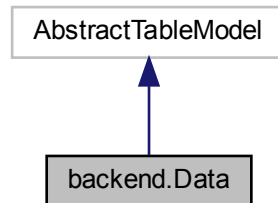
<i>w</i>	Az ablak szélessége.
<i>h</i>	Az ablak magassága.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

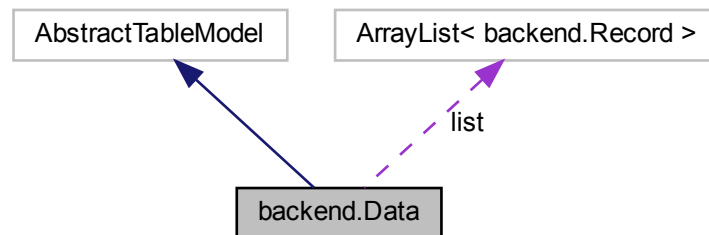
- src/backend/Controller.java

3.6. backend.Data osztályreferencia

A backend.Data osztály származási diagramja:



A backend.Data osztály együttműködési diagramja:



Publikus tagfüggvények

- int [getRowCount](#) ()
- int [getColumnCount](#) ()
- String [getColumnName](#) (int columnIndex)
- Object [getValueAt](#) (int rowIndex, int columnIndex)
- void [addRecord](#) (String playerName, [Time](#) time)

3.6.1. Részletes leírás

Adat osztály rekordok eltárolására és táblában való megjelenítésére.

3.6.2. Tagfüggvények dokumentációja

3.6.2.1. addRecord()

```
void backend.Data.addRecord (
    String playerName,
    Time time )
```

Rekord felvétele a táblába

Paraméterek

<i>playerName</i>	a játékos neve
<i>time</i>	a játékos ideje

3.6.2.2. getColumnCount()

```
int backend.Data.getColumnCount ( )
```

Visszatérési érték

Tábla oszlopainak számát adja vissza. (Név, idő)

3.6.2.3. getColumnName()

```
String backend.Data.getColumnName (
    int columnIndex )
```

Visszatérési érték

Tábla megfelelő oszlopának nevét adja vissza. Az első oszlop mindig a játékosok nevét, a második meg az idejüket tárolja.

3.6.2.4. getRowCount()

```
int backend.Data.getRowCount ( )
```

Visszatérési érték

Tábla sorainak számát adja vissza.

3.6.2.5. getValueAt()

```
Object backend.Data.getValueAt (
    int rowIndex,
    int columnIndex )
```

Érték megszerzése a tábla megfelelő sorából és oszlopából

Paraméterek

<i>rowIndex</i>	: A sor sorszáma
<i>columnIndex</i>	: Az oszlop sorszáma

Visszatérési érték

A tábla megfelelő cellájának értékét adja vissza.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `src/backend/Data.java`

3.7. backend.Difficulty felsoroló referencia

Publikus tagfüggvények

- void `setCustom` (int r, int c, int b)
- int `rows` ()
- int `cols` ()
- int `bombs` ()

Publikus attribútumok

- `EASY` =(8,8,10)
- `NORMAL` =(16,16,40)
- `HARD` =(16,30,99)
- `CUSTOM` =(0,0,0)
- int `cols`
- int `bombs`

3.7.1. Részletes leírás

A játék nehézségeit felsorakoztató enumeráció. Minden nehézségnek van egy adott sor- és oszlopszáma, illetve bombaszáma, ami legalább a 10%-a a tábla méretének.

3.7.2. Tagfüggvények dokumentációja

3.7.2.1. bombs()

```
int backend.Difficulty.bombs ( )
```

Get függvény a bombák lekérdezésére.

3.7.2.2. cols()

```
int backend.Difficulty.cols ( )
```

Get függvény az oszlopok lekérdezésére.

3.7.2.3. rows()

```
int backend.Difficulty.rows ( )
```

Get függvény a sorok lekérdezésére.

3.7.2.4. setCustom()

```
void backend.Difficulty.setCustom (
    int r,
    int c,
    int b )
```

Az egyéni nehézség beállítása.

Paraméterek

<i>r</i>	a nehézség sorainak száma.
<i>c</i>	a nehézség oszlopainak száma.
<i>b</i>	a nehézség bombáinak száma.

3.7.3. Adattagok dokumentációja

3.7.3.1. EASY

```
backend.Difficulty.EASY = (8, 8, 10)
```

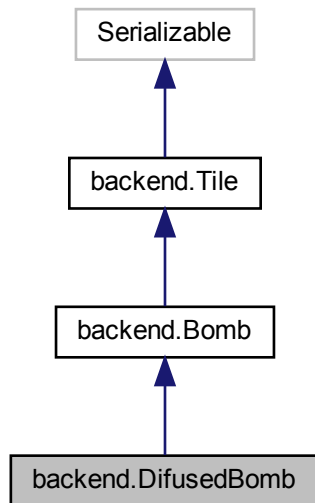
3 alapnehézség és egy egyedi nehézség van.

A dokumentáció ehhez az enum-hoz a következő fájl alapján készült:

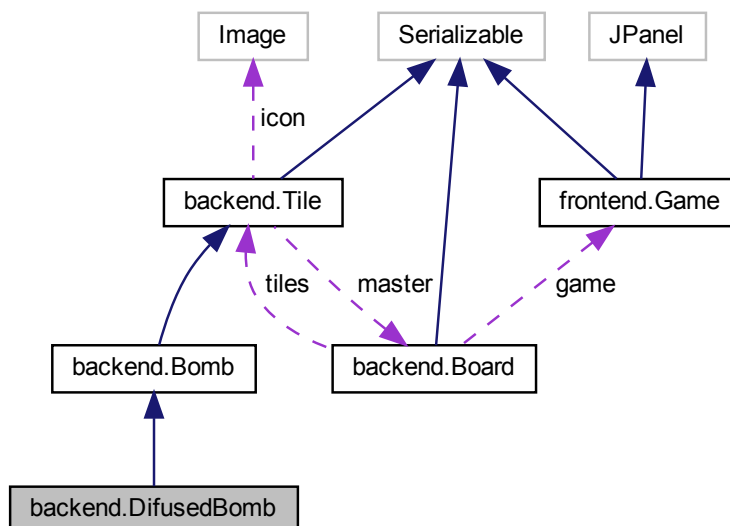
- src/backend/Difficulty.java

3.8. backend.DifusedBomb osztályreferencia

A backend.DifusedBomb osztály származási diagramja:



A backend.DifusedBomb osztály együttműködési diagramja:



Publikus tagfüggvények

- `DifusedBomb` (`Board b`)
- `void reveal ()`

Védett tagfüggvények

- `void changelcon ()`

További örökölt tagok

3.8.1. Részletes leírás

Hatástalanított bomba osztály. Bombának számít, viszont felfedéskor nem ér véget a játék.

3.8.2. Konstruktorok és destruktorok dokumentációja

3.8.2.1. `DifusedBomb()`

```
backend.DifusedBomb.DifusedBomb (
    Board b )
```

A létrehozása azonos a sima bombáéval.

Paraméterek

<code>b,az</code>	őzt tartalmazó tábla.
-------------------	-----------------------

3.8.3. Tagfüggvények dokumentációja

3.8.3.1. `changelcon()`

```
void backend.DifusedBomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: `backend.Bomb`.

3.8.3.2. reveal()

```
void backend.DifusedBomb.reveal ( )
```

Felfedés esetén ha még nincs felfedve, felfedi, és a bombák és zászlók számát csökkenti.

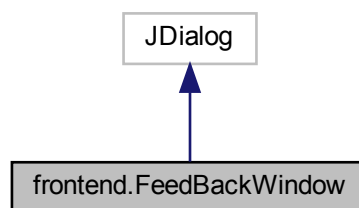
Újrimplementált ősök: [backend.Bomb](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

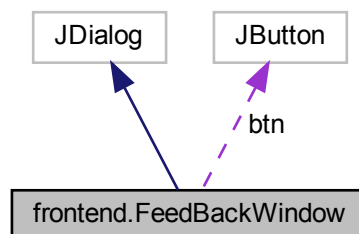
- `src/backend/DifusedBomb.java`

3.9. frontend.FeedBackWindow osztályreferencia

A frontend.FeedBackWindow osztály származási diagramja:



A frontend.FeedBackWindow osztály együttműködési diagramja:



Publikus tagfüggvények

- [FeedBackWindow](#) (String text, boolean success)

3.9.1. Részletes leírás

Visszajelzésért felelős ablak osztály. Üzenetet ad egy művelet sikerességéről.

3.9.2. Konstruktorok és destruktorok dokumentációja

3.9.2.1. FeedBackWindow()

```
frontend.FeedBackWindow.FeedBackWindow (
    String text,
    boolean success )
```

Az osztály konstruktora. Létrehozza a gombot és a hibaüzenet feliratát, illetve beállítja a grafikai részeket.

Paraméterek

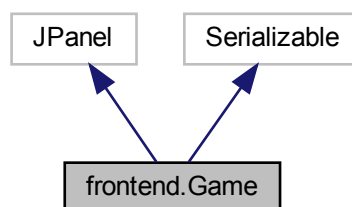
<i>text</i>	A hibaüzenet. Ez jelenik meg középen.
<i>success</i>	Jelzi, hogy a művelet sikerességét vagy hibáját jelzi-e az ablak.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

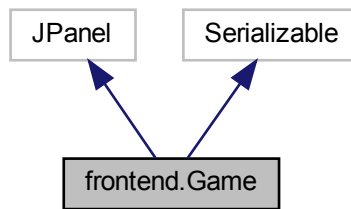
- src/frontend/FeedBackWindow.java

3.10. frontend.Game osztályreferencia

A frontend.Game osztály származási diagramja:



A frontend.Game osztály együttműködési diagramja:



Publikus tagfüggvények

- `Game` (`Controller` controller)
- `Time` `getTime` ()
- `void` `init` ()
- `void` `reload` (`Time` t)
- `boolean` `running` ()
- `boolean` `finished` ()
- `boolean` `started` ()
- `void` `setStarted` (`boolean` b)
- `Board` `getBoard` ()
- `void` `setFinished` (`boolean` b)
- `void` `setRunning` (`boolean` b)
- `void` `restart` ()
- `void` `start` ()
- `void` `paintComponent` (`Graphics` g)

3.10.1. Részletes leírás

Az aktuálisan futó játék osztálya. Méri a játékos idejét, illetve tárolja a játéktáblát.

3.10.2. Konstruktorkok és destruktorkok dokumentációja

3.10.2.1. Game()

```
frontend.Game.Game (
    Controller controller )
```

A játék konstruktora. Létrehozza a táblát az azt irányító controllerrel, beállítja a kezdeti értékeket és aktiválja a fejléc `Game` opcióját.

Paraméterek

<i>controller</i>	A táblát (is) irányító controller.
-------------------	------------------------------------

3.10.3. Tagfüggvények dokumentációja

3.10.3.1. finished()

```
boolean frontend.Game.finished ( )
```

Visszatérési érték

Visszaadja, hogy befejeződött-e a játék.

3.10.3.2. getBoard()

```
Board frontend.Game.getBoard ( )
```

Visszatérési érték

visszaadja a játéktáblát

3.10.3.3. getTime()

```
Time frontend.Game.getTime ( )
```

Visszaadja a játékos idejét.

3.10.3.4. init()

```
void frontend.Game.init ( )
```

A kezdeti értékek beállítása. alaptól a játék nem fut, nem fejeződött be és nem is kezdődött el.

3.10.3.5. paintComponent()

```
void frontend.Game.paintComponent (
    Graphics g )
```

A grafikus elemek kirajzolása

3.10.3.6. reload()

```
void frontend.Game.reload (
    Time t )
```

A játék betöltésekor újratölti a képeket, beállítja az időzítőt a mentett játék idejére és elindítja az időmérő szálat.

3.10.3.7. restart()

```
void frontend.Game.restart ( )
```

Újrakezdi a játékot. Új játék indításakor használta.

3.10.3.8. running()

```
boolean frontend.Game.running ( )
```

Visszatérési érték

Visszaadja, hogy fut-e a játék.

3.10.3.9. setFinished()

```
void frontend.Game.setFinished (
    boolean b )
```

Beállítja, hogy befejeződött-e a játék

Paraméterek

A	beállítandó érték.
---	--------------------

3.10.3.10. setRunning()

```
void frontend.Game.setRunning (
    boolean b )
```

Beállítja, hogy fut-e a játék

Paraméterek

A	beállítandó érték.
---	--------------------

3.10.3.11. setStarted()

```
void frontend.Game.setStarted (
    boolean b )
```

Beállítja, hogy elkezdődött-e a játék.

Paraméterek

<i>b</i>	A beállítandó érték.
----------	----------------------

3.10.3.12. start()

```
void frontend.Game.start ( )
```

Elindítja a játékot és a stoppert az első kattintás után. Ekkor generálódnak a bombák és számítnak ki a cellák körüli bombák számai.

3.10.3.13. started()

```
boolean frontend.Game.started ( )
```

Visszatérési érték

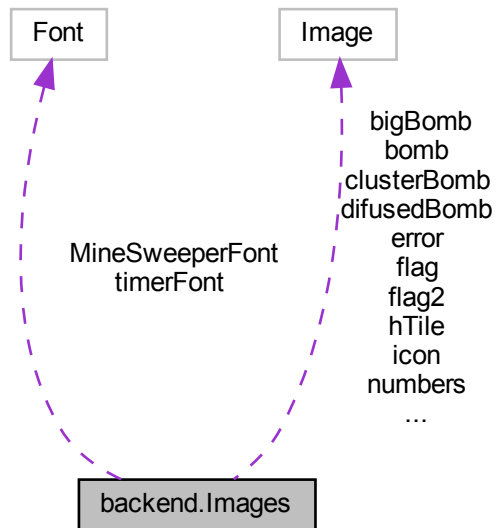
Visszaadja, hogy elkezdődött-e a játék.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- src/frontend/Game.java

3.11. backend.Images osztályreferencia

A backend.Images osztály együttműködési diagramja:



Statikus publikus tagfüggvények

- static void `loadImages` () throws Exception

Statikus publikus attribútumok

- static Font `timerFont`
- static Font `MineSweeperFont`
- static Image `numbers` []
- static Image `unknown`
- static Image `bomb`
- static Image `resetFlagBomb`
- static Image `redBomb`
- static Image `difusedBomb`
- static Image `resetBomb`
- static Image `bigBomb`
- static Image `clusterBomb`
- static Image `flag`
- static Image `flag2`
- static Image `hTile`
- static Image `icon`
- static Image `error`
- static Image `success`

3.11.1. Részletes leírás

`Images` osztály, ami a játék képeit, betűtípusait tartalmazza.

3.11.2. Tagfüggvények dokumentációja

3.11.2.1. `loadImages()`

```
static void backend.Images.loadImages ( ) throws Exception [static]
```

Betölti a játék elején a megfelelő ikonokat.

Kivételek

<i>Exception</i>	ha nem sikerült legalább egy ikont betölteni, hibaüzenetet kapunk.
------------------	--

3.11.3. Adattagok dokumentációja

3.11.3.1. `bigBomb`

```
Image backend.Images.bigBomb [static]
```

A nagy bomba ikonja

3.11.3.2. `bomb`

```
Image backend.Images.bomb [static]
```

A sima bomba ikonja

3.11.3.3. `clusterBomb`

```
Image backend.Images.clusterBomb [static]
```

A cluster bomba ikonja

3.11.3.4. `difusedBomb`

```
Image backend.Images.difusedBomb [static]
```

A hatástalanított bomba ikonja

3.11.3.5. error

`Image backend.Images.error [static]`

A hibaüzenetek ikonja

3.11.3.6. flag

`Image backend.Images.flag [static]`

A sima zászló ikonja

3.11.3.7. flag2

`Image backend.Images.flag2 [static]`

Az áthúzott zászló ikonja

3.11.3.8. hTile

`Image backend.Images.hTile [static]`

A fedett cella ikonja

3.11.3.9. icon

`Image backend.Images.icon [static]`

A játék ablakának ikonja

3.11.3.10. MineSweeperFont

`Font backend.Images.MineSweeperFont [static]`

A játék betűtípusa

3.11.3.11. numbers

`Image backend.Images.numbers[] [static]`

A cellák körül lévő bombák számainak ikonjai

3.11.3.12. redBomb

`Image backend.Images.redBomb [static]`

A piros hátterű bomba ikonja.

3.11.3.13. resetBomb

```
Image backend.Images.resetBomb [static]
```

A pályát újrakezdő bomba ikonja

3.11.3.14. resetFlagBomb

```
Image backend.Images.resetFlagBomb [static]
```

A zászó eltüntető bomba ikonja

3.11.3.15. success

```
Image backend.Images.success [static]
```

A sikeres műveletek ikonja

3.11.3.16. timerFont

```
Font backend.Images.timerFont [static]
```

Az időzítő betűtípusa

3.11.3.17. unknown

```
Image backend.Images.unknown [static]
```

A cellák körül lévő (8-nál több) bombák ikonja

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `src/backend/Images.java`

3.12. backend.Main osztályreferencia

Statikus publikus tagfüggvények

- `static void main (String[] args)`

3.12.1. Részletes leírás

A main osztály. Innen indul a program.

3.12.2. Tagfüggvények dokumentációja

3.12.2.1. main()

```
static void backend.Main.main (
    String[] args ) [static]
```

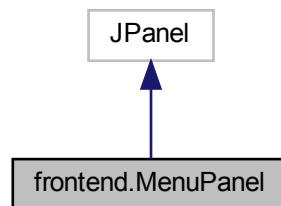
Ha sikeresen beöltötte a képeket, akkor megjelenik a program ablaka. Ha nem, akkor hibaüzenetet kapunk.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

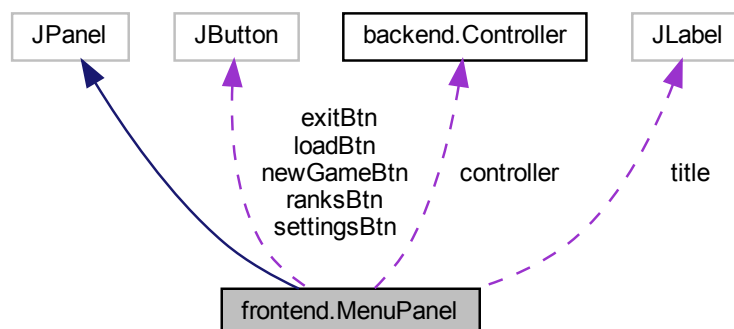
- src/backend/Main.java

3.13. frontend.MenuPanel osztályreferencia

A frontend.MenuPanel osztály származási diagramja:



A frontend.MenuPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- [MenuPanel](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [setActionListeners](#) ()

3.13.1. Részletes leírás

A főmenüt megjelenítő panel. Innen át lehet térni a legtöbb nézetbe.

3.13.2. Konstruktorkok és destruktorkok dokumentációja

3.13.2.1. MenuPanel()

```
frontend.MenuPanel.MenuPanel (
    Controller controller )
```

A panel konstruktora. Beállítja a controllert, létrehozza a grafikus elemeket és beállítja azok actionListener-jeit.

Paraméterek

<i>controller</i>	Az őt (is) irányító controller.
-------------------	---------------------------------

3.13.3. Tagfüggvények dokumentációja

3.13.3.1. initComponents()

```
void frontend.MenuPanel.initComponents ( )
```

A grafikus elemek létrehozása.

3.13.3.2. setActionListeners()

```
void frontend.MenuPanel.setActionListeners ( )
```

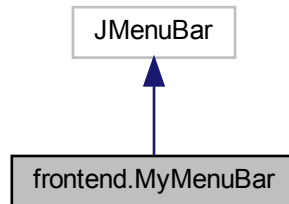
Az actionListener-ek beállítása.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

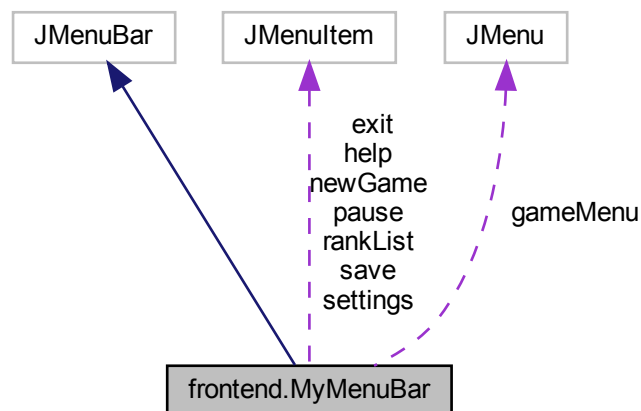
- src/frontend/MenuPanel.java

3.14. frontend.MyMenuBar osztályreferencia

A frontend.MyMenuBar osztály származási diagramja:



A frontend.MyMenuBar osztály együttműködési diagramja:



Publikus tagfüggvények

- `MyMenuBar` (`Controller` controller)
- void `initComponents` ()
- void `setGameBarEn` (boolean b)
- void `setSaveBtnEn` (boolean b)
- void `setPauseBtnText` (boolean running)
- void `setActionListeners` ()

3.14.1. Részletes leírás

A játék fejlécét reprezentáló saját készítésű `JMenuBar`.

3.14.2. Konstruktorkok és destruktorkok dokumentációja

3.14.2.1. MyMenuBar()

```
frontend.MyMenuBar.MyMenuBar (
    Controller controller )
```

A menü fejléc konstruktora. Beállítja a controllert, létrehozza az opcióit, majd beállítja azok actionListener-jeit.

Paraméterek

<i>controller</i>	Az őt (is) irányító controller.
-------------------	---------------------------------

3.14.3. Tagfüggvények dokumentációja

3.14.3.1. initComponents()

```
void frontend.MyMenuBar.initComponents ( )
```

A grafikus elemek létrehozása.

3.14.3.2. setActionListeners()

```
void frontend.MyMenuBar.setActionListeners ( )
```

Az actionListener-ek beállítása. A help opció egy weboldalt nyit meg, ahol a játék alapvető logikáját tanulhatják meg.

3.14.3.3. setGameBarEn()

```
void frontend.MyMenuBar.setGameBarEn (
    boolean b )
```

A "Game" nevű opció engedélyezése.

Paraméterek

<i>b</i>	Az engedélyezés logikai értéke.
----------	---------------------------------

3.14.3.4. setPauseBtnText()

```
void frontend.MyMenuBar.setPauseBtnText (
    boolean running )
```

A mentés gomb szövegének beállítása

Paraméterek

<i>running</i>	Annak a logikai értéke, hogy jelenleg fut-e a játék.
----------------	--

3.14.3.5. setSaveBtnEn()

```
void frontend.MyMenuBar.setSaveBtnEn (
    boolean b )
```

A mentés gomb engedélyezése. Ha véget ért a játék vagy nyert a játékos, akkor nem lehet menteni már.

Paraméterek

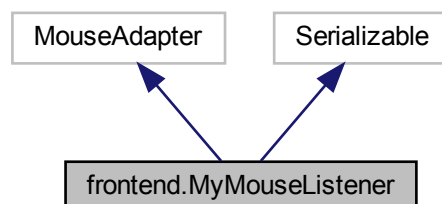
<i>b</i>	Az engedélyezés logikai értéke.
----------	---------------------------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

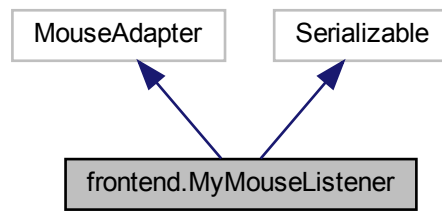
- src/frontend/MyMenuBar.java

3.15. frontend.MyMouseListener osztályreferencia

A frontend.MyMouseListener osztály származási diagramja:



A frontend.MyMouseListener osztály együttműködési diagramja:



Publikus tagfüggvények

- `MouseListener (Game g)`
- void `setBoard (Board b)`
- void `mouseReleased (MouseEvent e)`

3.15.1. Részletes leírás

Egérkezelő osztály. Ez felel a mezők felfedésének, megjelölésének detektálásáért.

3.15.2. Konstruktorok és destruktorok dokumentációja

3.15.2.1. MyMouseListener()

```
frontend.MyMouseListener.MyMouseListener (
    Game g )
```

Az egérkezelő konstruktora. Beállítja a játékot és a táblát.

3.15.3. Tagfüggvények dokumentációja

3.15.3.1. mouseReleased()

```
void frontend.MyMouseListener.mouseReleased (
    MouseEvent e )
```

Az egér felengedése. Ha felengedi a játékos az egér egy gombját, akkor kiszámolja a koordinátákra eső sor- és oszlopszámot, majd bal kattintás esetén felfedi a cellát (és esetleg elindítja a játék futását), jobb kattintás esetén megjelöli a cellát (ha fut a játék).

3.15.3.2. setBoard()

```
void frontend.MyMouseListener.setBoard (
    Board b )
```

Beállítja a táblát, amit figyel.

Paraméterek

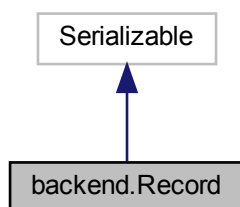
<i>b</i>	A beállítandó tábla.
----------	----------------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

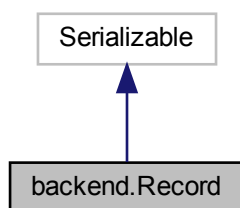
- src/frontend/MyMouseListener.java

3.16. backend.Record osztályreferencia

A backend.Record osztály származási diagramja:



A backend.Record osztály együttműködési diagramja:



Publikus tagfüggvények

- `Record` (String name, Time t)
- String `getName` ()
- Time `getTime` ()

3.16.1. Részletes leírás

Egy adatrekordot reprezentáló osztály. Tartalmazza a játékos nevét és idejét.

3.16.2. Konstruktorok és destruktorok dokumentációja

3.16.2.1. Record()

```
backend.Record.Record (
    String name,
    Time t )
```

A rekord konstruktora. Beállítja a nevet és az időt.

Paraméterek

<i>name</i>	a játékos neve
<i>t</i>	a játékos ideje

3.16.3. Tagfüggvények dokumentációja

3.16.3.1. getName()

```
String backend.Record.getName ( )
```

Get függvény a játékos nevének lekérdezésére.

Visszatérési érték

a játékos neve

3.16.3.2. getTime()

```
Time backend.Record.getTime ( )
```

Get függvény a játékos idejének lekérdezésére.

Visszatérési érték

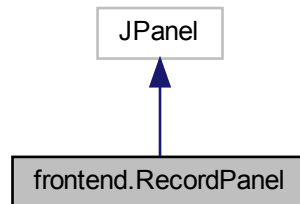
a játékos ideje

Ez a dokumentáció az osztályról a következő fájl alapján készült:

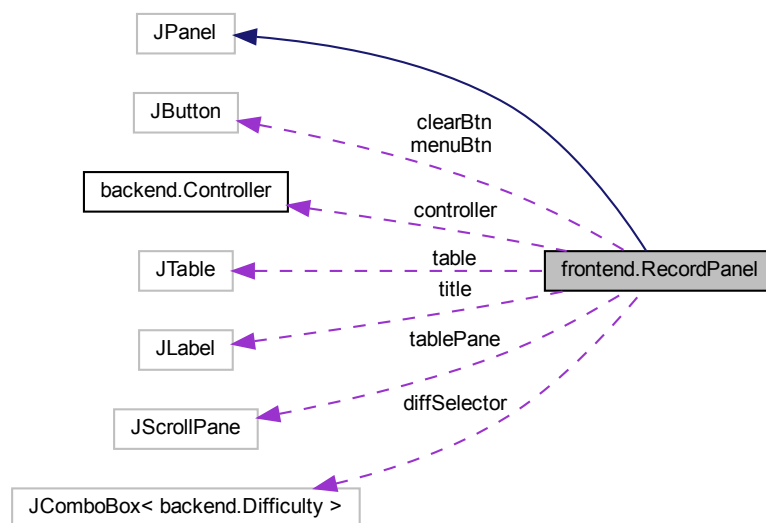
- src/backend/Record.java

3.17. frontend.RecordPanel osztályreferencia

A frontend.RecordPanel osztály származási diagramja:



A frontend.RecordPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- [RecordPanel](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [setActionListeners](#) ()

3.17.1. Részletes leírás

A rekordok megtekintésére szolgáló JPanel. 3 nehézség között lehet váltani, mindegyiket egy táblázatban jeleníti meg.

3.17.2. Konstruktorok és destruktorok dokumentációja

3.17.2.1. RecordPanel()

```
frontend.RecordPanel.RecordPanel (
    Controller controller )
```

A panel konstruktora. Beállítja a controllert, a megjelenítendő (alap) adatot, létrehozza a grafikus elemeket, majd beállítja azok ActionListener-jeit.

Paraméterek

<i>controller</i>	Az őt (is) irányító controller.
-------------------	---------------------------------

3.17.3. Tagfüggvények dokumentációja

3.17.3.1. initComponents()

```
void frontend.RecordPanel.initComponents ( )
```

A grafikus elemek létrehozása.

3.17.3.2. setActionListeners()

```
void frontend.RecordPanel.setActionListeners ( )
```

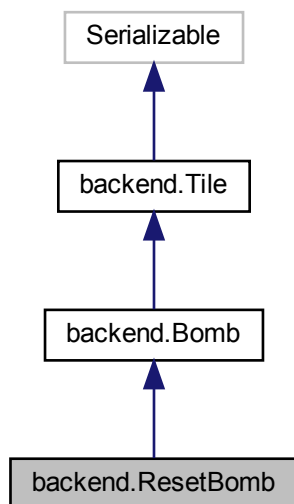
Az ActionListener-ek beállítása.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

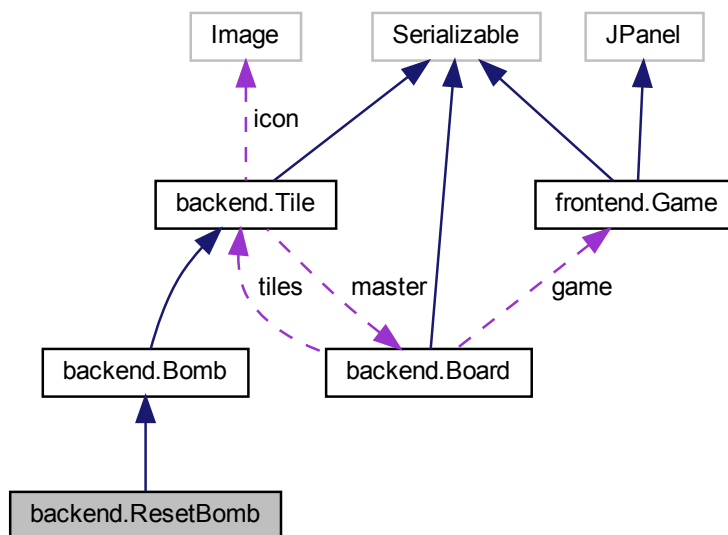
- src/frontend/RecordPanel.java

3.18. backend.ResetBomb osztályreferencia

A backend.ResetBomb osztály származási diagramja:



A backend.ResetBomb osztály együttműködési diagramja:



Publikus tagfüggvények

- [ResetBomb](#) ([Board](#) b)

Védett tagfüggvények

- void [changeIcon](#) ()
- void [execute](#) ()

További örökölt tagok

3.18.1. Részletes leírás

Pályát újrakezdő bomba. Ha felfedik, újragenerálja a pályát, viszont az időzítő halad tovább.

3.18.2. Konstruktorok és destruktorok dokumentációja

3.18.2.1. ResetBomb()

```
backend.ResetBomb.ResetBomb (
    Board b )
```

A létrehozása azonos a sima bombáéval.

Paraméterek

<i>b, az</i>	ő t tartalmazó tábla.
--------------	-----------------------

3.18.3. Tagfüggvények dokumentációja

3.18.3.1. changeIcon()

```
void backend.ResetBomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: [backend.Bomb](#).

3.18.3.2. execute()

```
void backend.ResetBomb.execute ( ) [protected]
```

Aktiválásakor újraindítja a táblát és a játékot.

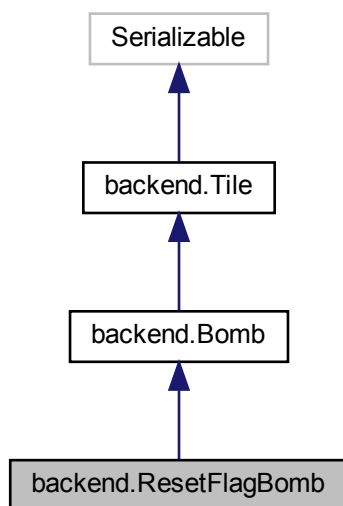
Újrimplementált ősök: [backend.Bomb](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

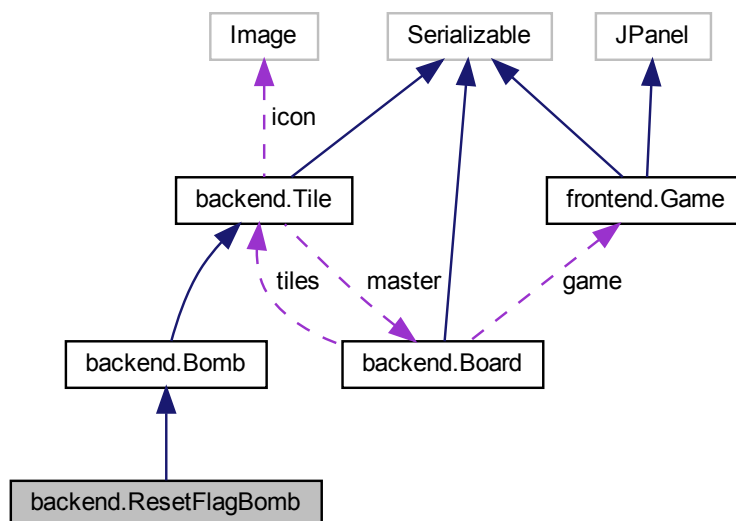
- src/backend/ResetBomb.java

3.19. backend.ResetFlagBomb osztályreferencia

A backend.ResetFlagBomb osztály származási diagramja:



A backend.ResetFlagBomb osztály együttműködési diagramja:



Publikus tagfüggvények

- [ResetFlagBomb](#) ([Board](#) b)

Védett tagfüggvények

- void `changeIcon` ()
- void `execute` ()

További örökölt tagok

3.19.1. Részletes leírás

Lerakott zászlókat eltüntető bomba. Ha felfedik, a már elhelyezett zászlókat leszedi a pályáról.

3.19.2. Konstruktorok és destruktorok dokumentációja

3.19.2.1. ResetFlagBomb()

```
backend.ResetFlagBomb.ResetFlagBomb (
    Board b )
```

A létrehozása azonos a sima bombáéval.

Paraméterek

<code>b,az</code>	őit tartalmazó tábla.
-------------------	-----------------------

3.19.3. Tagfüggvények dokumentációja

3.19.3.1. changeIcon()

```
void backend.ResetFlagBomb.changeIcon ( ) [protected]
```

A felfedéskor megváltozik az ikonja fedett mezőről a neki megfelelőre.

Újrimplementált ősök: `backend.Bomb`.

3.19.3.2. execute()

```
void backend.ResetFlagBomb.execute ( ) [protected]
```

Végrehajtáskor meghívja a tábla zászló eltüntető függvényét.

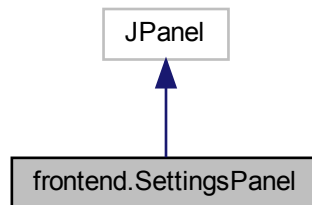
Újrimplementált ősök: `backend.Bomb`.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

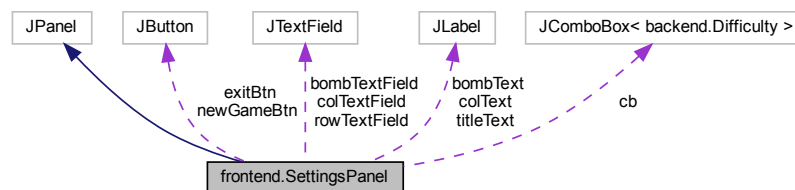
- `src/backend/ResetFlagBomb.java`

3.20. frontend.SettingsPanel osztályreferencia

A frontend.SettingsPanel osztály származási diagramja:



A frontend.SettingsPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- [SettingsPanel](#) ([Controller](#) controller)
- boolean [submit](#) ()
- void [initComponents](#) ()
- void [setActionListeners](#) ()

3.20.1. Részletes leírás

A beállítások panel. Itt lehet a játék nehézségét beállítani.

3.20.2. Konstruktorok és destruktorok dokumentációja

3.20.2.1. SettingsPanel()

```

frontend.SettingsPanel.SettingsPanel (
    Controller controller )
  
```

A panel konstruktora. Beállítja a controllert, majd beállítja a grafikus elemeket és beállítja azok actionListener-jeit.

Paraméterek

<i>controller</i>	Az őt (is) irányító controller.
-------------------	---------------------------------

3.20.3. Tagfüggvények dokumentációja

3.20.3.1. initComponents()

```
void frontend.SettingsPanel.initComponents ( )
```

A grafikus elemek létrehozása.

3.20.3.2. setActionListeners()

```
void frontend.SettingsPanel.setActionListeners ( )
```

ActionListener-ek beállítása.

3.20.3.3. submit()

```
boolean frontend.SettingsPanel.submit ( )
```

A beállítások elmentése. Ha egyéni nehézséget állítunk be és rosszak az értékei, akkor hibaüzenetet ad és nem hajtja végre a mentést.

Visszatérési érték

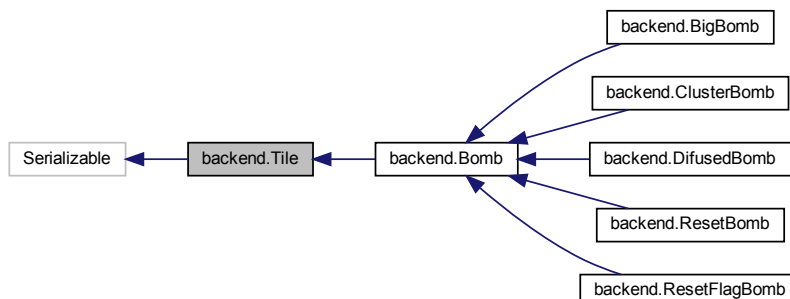
A mentés sikeressége.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

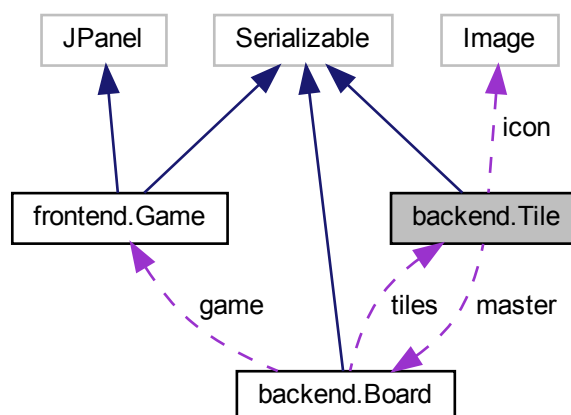
- src/frontend/SettingsPanel.java

3.21. backend.Tile osztályreferencia

A backend.Tile osztály származási diagramja:



A backend.Tile osztály együttműködési diagramja:



Publikus tagfüggvények

- [Tile](#) ([Board](#) b)
- [int](#) [getValue](#) ()
- [int](#) [getBombsAround](#) ()
- [boolean](#) [isFlagged](#) ()
- [void](#) [setFlagged](#) ([boolean](#) b)
- [void](#) [setBombsAround](#) ([int](#) b)
- [void](#) [setRevealed](#) ([boolean](#) b)
- [void](#) [setCoords](#) ([int](#) x, [int](#) y)
- [void](#) [flag](#) ()

- void `loadIcon` ()
- Image `getCorrespondingIcon` ()
- void `reveal` ()
- void `revealEnd` ()
- void `paintComponent` (Graphics g)

Statikus publikus tagfüggvények

- static int `getW` ()

Védett tagfüggvények

- void `changeIcon` ()

Védett attribútumok

- boolean `isFlagged`
- boolean `isRevealed`
- `Board` master
- int x
- int `bombsAround`
- transient Image `icon`

Statikus védett attribútumok

- static int `w` = 25

3.21.1. Részletes leírás

Cella ősosztály. A táblát ezen cellák alkotják, melyek lehetnek sima cellák, illetve bombák.

3.21.2. Konstruktorok és destruktorok dokumentációja

3.21.2.1. `Tile()`

```
backend.Tile.Tile (
    Board b )
```

Létrehozásakor beállítja, hogy nincs megjelölve, illetve felfedve. Alapból egy fedett mező az ikonja, beállítja az őt tartalmazó táblát.

Paraméterek

<code>b</code>	: az őt tartalmazó tábla.
----------------	---------------------------

3.21.3. Tagfüggvények dokumentációja

3.21.3.1. changelcon()

```
void backend.Tile.changeIcon ( ) [protected]
```

Átállítja az ikont a megfelelő ikonjára.

Újrimplementáló leszármazottak: [backend.ResetFlagBomb](#), [backend.ResetBomb](#), [backend.DifusedBomb](#), [backend.ClusterBomb](#), [backend.Bomb](#) és [backend.BigBomb](#).

3.21.3.2. flag()

```
void backend.Tile.flag ( )
```

Cella megjelölése. Ha még van elég elhelyezhető zászló, akkor megcseréli a státuszát, illetve ennek megfelelően átírja a tábla zászlóinak számát. Ha nincs elég zászló már, akkor ha meg van jelölve, leszedi róla a jelölést. Végezetül megváltoztatja az ikonját annak függvényében, hogy most meg van-e jelölve vagy sem.

Újrimplementáló leszármazottak: [backend.Bomb](#).

3.21.3.3. getBombsAround()

```
int backend.Tile.getBombsAround ( )
```

Get függvény a cella körüli bombák számának lekérdezésére.

3.21.3.4. getCorrespondingIcon()

```
Image backend.Tile.getCorrespondingIcon ( )
```

A megfelelő ikont adja vissza a bombaszámtól függően. Ha több, mint 8 bomba van a cella körül, akkor egy "?" ikont állít be.

Visszatérési érték

a megfelelő ikon.

3.21.3.5. `getValue()`

```
int backend.Tile.getValue ( )
```

A `getBombsAround` függvénynél használatos.

Visszatérési érték

0 bombának megfelelő az értéke.

Újrimplementáló leszármazottak: [backend.Bomb](#) és [backend.BigBomb](#).

3.21.3.6. `getW()`

```
static int backend.Tile.getW ( ) [static]
```

Statikus `get` függvény a cella szélességének lekérdezésére.

3.21.3.7. `isFlagged()`

```
boolean backend.Tile.isFlagged ( )
```

Meg van-e jelölve a cella

3.21.3.8. `loadIcon()`

```
void backend.Tile.loadIcon ( )
```

Újratölti a cella ikonját. (Játék betöltésekor hasznos) Ha nincs felfedve, akkor ha meg van jelölve zászló, egyéb esetben sima cella lesz az ikonja. Ha fel van fedve, akkor a [changelcon\(\)](#) függvényt hívja meg.

3.21.3.9. `paintComponent()`

```
void backend.Tile.paintComponent (
    Graphics g )
```

Cella kirajzolása. Egy négyzetes képet rajzol a cella megfelelő koordinátáira.

Paraméterek

<i>g</i>	: A grafikáért felelős paraméter. Az őt tároló Board adja át.
----------	---

3.21.3.10. reveal()

```
void backend.Tile.reveal ( )
```

Cella felfedése. Elsősorban csökketi a tábla fedett mezőinek számát, aztán beállítja a megfelelő ikonját, valamint hogy fel van fedve már.

Újrimplementáló leszármazottak: [backend.DifusedBomb](#) és [backend.Bomb](#).

3.21.3.11. revealEnd()

```
void backend.Tile.revealEnd ( )
```

Végző felfedés. A játék végénél használatos, amikor a bombák miatt felrobban a pálya. Először csökkenti a fedett mezők számát, majd beállítja a megfelelő ikont. Ha meg volt jelölve egy sima cella, akkor egy áthúzott zászlót állít be ikonnak.

Újrimplementáló leszármazottak: [backend.Bomb](#).

3.21.3.12. setBombsAround()

```
void backend.Tile.setBombsAround (
    int b )
```

Beállítja a cella körüli bombák számát.

Újrimplementáló leszármazottak: [backend.Bomb](#).

3.21.3.13. setCoords()

```
void backend.Tile.setCoords (
    int x,
    int y )
```

Beállítja a cella koordinátáit a képernyőn.

Paraméterek

<i>x</i>	
<i>y</i>	

3.21.3.14. setFlagged()

```
void backend.Tile.setFlagged (
    boolean b )
```

Beállítja, hogy meg van-e jelölve a cella. A resetFlags-nél használatos

3.21.3.15. setRevealed()

```
void backend.Tile.setRevealed (
    boolean b )
```

Beállítja, hogy fel van-e fedve a cella. Legfőképp a pálya újratekésztésénél van haszna.

Paraméterek

<i>b</i>	
----------	--

3.21.4. Adattagok dokumentációja

3.21.4.1. bombsAround

```
int backend.Tile.bombsAround [protected]
```

A körülötte lévő bombák száma.

3.21.4.2. icon

```
transient Image backend.Tile.icon [protected]
```

A megjelenítendő ikonja. A játék menete során változik.

3.21.4.3. isFlagged

```
boolean backend.Tile.isFlagged [protected]
```

Meg van-e jelölve zászlóval. Ilyenkor nem lehet felfedni.

3.21.4.4. isRevealed

```
boolean backend.Tile.isRevealed [protected]
```

Fel van-e fedve már. Azért kell, hogy a felfedés flood-fill algoritmus megálljon.

3.21.4.5. master

```
Board backend.Tile.master [protected]
```

Az őt tartalmazó tábla.

3.21.4.6. w

```
int backend.Tile.w = 25 [static], [protected]
```

A cellák szélessége (és egyben magassága)

3.21.4.7. x

```
int backend.Tile.x [protected]
```

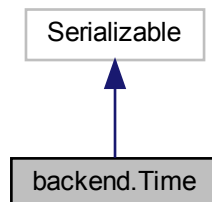
A cella x és y koordinátája a képernyőn.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

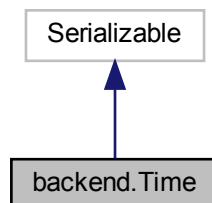
- src/backend/Tile.java

3.22. backend.Time osztályreferencia

A backend.Time osztály származási diagramja:



A backend.Time osztály együttműködési diagramja:



Publikus tagfüggvények

- `Time` ()
- void `increase` ()
- boolean `lt` (int num)
- int `getM` ()
- int `getS` ()
- String `toString` ()
- int `compareTo` (`Time` time)

3.22.1. Részletes leírás

Egyszerű idő osztály, ami a mostani feladat megoldásához elegendő.

3.22.2. Konstruktorkok és destruktorkok dokumentációja

3.22.2.1. `Time()`

```
backend.Time.Time ( )
```

Az idő konstruktora, beállítja a percet és a másodpercet 0-ra.

3.22.3. Tagfüggvények dokumentációja

3.22.3.1. `compareTo()`

```
int backend.Time.compareTo (  
    Time time )
```

Sorbarendezéshez használható összehasonlítás

Paraméterek

<i>time</i>	egy másik idő
-------------	---------------

Visszatérési érték

negatív érték, ha ez az idő kisebb a paraméterben megadottnál.

3.22.3.2. getM()

```
int backend.Time.getM ( )
```

Get függvény a percek lekérésére.

Visszatérési érték

A percek értéke.

3.22.3.3. getS()

```
int backend.Time.getS ( )
```

Get függvény a másodpercek lekérésére.

Visszatérési érték

A másodpercek értéke.

3.22.3.4. increase()

```
void backend.Time.increase ( )
```

Megnöveli eggyel az idő értékét. Ha a másodperc elér a 60-at, akkor a percet növeli eggyel, azt meg lenullázza.

3.22.3.5. lt()

```
boolean backend.Time.lt (
    int num )
```

Függvény annak eldöntésére, hogy az idő kisebb-e, mint a megadott érték.

Paraméterek

<i>num</i>	A vizsgált idő másodpercekben
------------	-------------------------------

Visszatérési érték

true, ha valóban kisebb.

3.22.3.6. toString()

```
String backend.Time.toString ( )
```

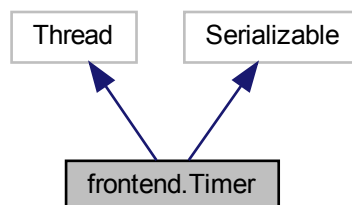
Kiíratáshoz használható [toString\(\)](#) metódus.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

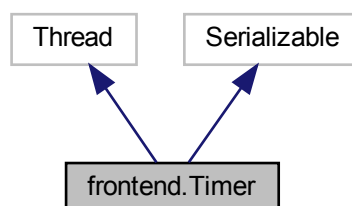
- src/backend/Time.java

3.23. frontend.Timer osztályreferencia

A frontend.Timer osztály származási diagramja:



A frontend.Timer osztály együttműködési diagramja:



Publikus tagfüggvények

- `Timer ()`
- `Timer (Time t)`
- `int getHeight ()`
- `boolean started ()`
- `String toString ()`
- `Time getTime ()`
- `void start ()`
- `void run ()`
- `void setFinished (boolean b)`
- `void setRunning (boolean b)`
- `boolean running ()`
- `void paintComponent (Graphics g, int width)`

3.23.1. Részletes leírás

Idő számlálására alkalmas osztály. 00:00-tól elkezdve számolja a játékos idejét. Külön szálként működik.

3.23.2. Konstruktorok és destruktorok dokumentációja

3.23.2.1. `Timer()` [1/2]

```
frontend.Timer.Timer ( )
```

Az időszámláló osztály konstruktora. Létrehozza az időt, illetve beállítja a kezdeti értékeket.

3.23.2.2. `Timer()` [2/2]

```
frontend.Timer.Timer (
    Time t )
```

Egy másik konstruktor, ami játék betöltésekor használt.

Paraméterek

<code>t</code>	A már meglévő játék ideje.
----------------	----------------------------

3.23.3. Tagfüggvények dokumentációja

3.23.3.1. getHeight()

```
int frontend.Timer.getHeight ( )
```

Visszaadja az időzítő kirajzolásának magasságát. (A még elhelyezhető zászlók kirajzolásához kell.)

Visszatérési érték

A kirajzolás magassága.

3.23.3.2. getTime()

```
Time frontend.Timer.getTime ( )
```

Visszatérési érték

Visszaadja a mért időt.

3.23.3.3. paintComponent()

```
void frontend.Timer.paintComponent (
    Graphics g,
    int width )
```

Kirajzolja az időzítőt az ablak szélességének megfelelően bal oldalra.

Paraméterek

<i>g</i>	A grafikáért felelős paraméter.
<i>width</i>	Az ablak szélessége.

3.23.3.4. run()

```
void frontend.Timer.run ( )
```

A szál futása. Amíg nem fejeződött be, addig másodpercenként számol egyet.

3.23.3.5. running()

```
boolean frontend.Timer.running ( )
```

Visszaadja, hogy fut-e az időzítő.

3.23.3.6. setFinished()

```
void frontend.Timer.setFinished (
    boolean b )
```

Beállítja a finished mező értékét

Paraméterek

<i>b</i>	A beállítandó érték.
----------	----------------------

3.23.3.7. setRunning()

```
void frontend.Timer.setRunning (
    boolean b )
```

Beállítja a running mező értékét.

Paraméterek

<i>b</i>	A beállítandó érték.
----------	----------------------

3.23.3.8. start()

```
void frontend.Timer.start ( )
```

A szál elindítása. A started mező értékét igazra állítja.

3.23.3.9. started()

```
boolean frontend.Timer.started ( )
```

Visszatérési érték

Elkezdődött-e már a számláló.

3.23.3.10. toString()

```
String frontend.Timer.toString ( )
```

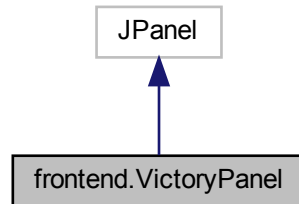
Kirajzoláshoz használt [toString\(\)](#) metódus. 2db 0-ra kerekíti az értékeket.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

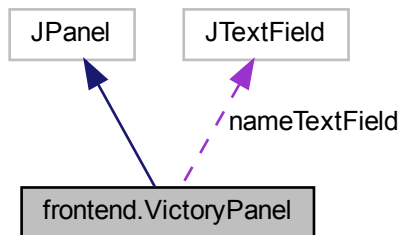
- src/frontend/Timer.java

3.24. frontend.VictoryPanel osztályreferencia

A frontend.VictoryPanel osztály származási diagramja:



A frontend.VictoryPanel osztály együttműködési diagramja:



Publikus tagfüggvények

- [VictoryPanel](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [setActionListeners](#) ()

3.24.1. Részletes leírás

A játék megnyerése során megjelenő nézet. Lehetőség nyílik elmenteni az időt, ha nem egyéni nehézségen játszottunk.

3.24.2. Konstruktorkok és destruktorkok dokumentációja

3.24.2.1. VictoryPanel()

```
frontend.VictoryPanel.VictoryPanel (
    Controller controller )
```

A panel konstruktora. Beállítja a controllert, lekéri a játék idejét, beállítja az egyéni nehézséget ellenőrző változó értékét. Ezt követően létrehozza a grafikus elemeket és beállítja azok ActionListener-jeit.

Paraméterek

<i>controller</i>	Az őt (is) irányító controller.
-------------------	---------------------------------

3.24.3. Tagfüggvények dokumentációja

3.24.3.1. initComponents()

```
void frontend.VictoryPanel.initComponents ( )
```

A grafikus elemek létrehozása.

3.24.3.2. setActionListeners()

```
void frontend.VictoryPanel.setActionListeners ( )
```

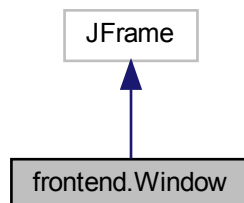
Az ActionListener-ek beállítása.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

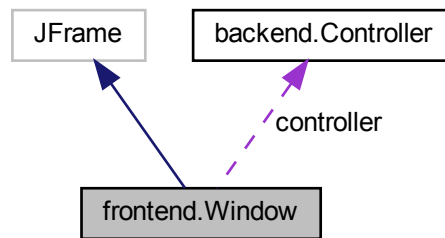
- src/frontend/VictoryPanel.java

3.25. frontend.Window osztályreferencia

A frontend.Window osztály származási diagramja:



A frontend.Window osztály együttműködési diagramja:



Publikus tagfüggvények

- [Window](#) (String title, [Controller](#) controller, Image img)
- void [initComponents](#) ()
- void [resetSize](#) ()
- void [setPanel](#) (JPanel panel)

3.25.1. Részletes leírás

A program ablaka. Erre helyezzük el a különböző nézetek paneljeit.

3.25.2. Konstruktorkok és destruktorkok dokumentációja

3.25.2.1. Window()

```
frontend.Window.Window (
    String title,
    Controller controller,
    Image img )
```

Az ablak konstruktora. Az alapértelmezett ablakméretet beállítja 600x500-asra, beállítja az ablak szükséges adatait, beállítja a kontrollert, betölti a mentett rekordokat, létrehozza a grafikus elemeket, majd megjeleníti az ablakot.

Paraméterek

<i>title</i>	Az ablak címe.
<i>controller</i>	Az őt (is) irányító controller.

3.25.3. Tagfüggvények dokumentációja

3.25.3.1. initComponents()

```
void frontend.Window.initComponents ( )
```

A grafikus elemek létrehozása.

3.25.3.2. resetSize()

```
void frontend.Window.resetSize ( )
```

Visszaállítja az ablak eredeti méretét.

3.25.3.3. setPanel()

```
void frontend.Window.setPanel (
    JPanel panel )
```

Beállítja az aktív panelt.

Paraméterek

<i>panel</i>	A beállítandó panel.
--------------	----------------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- src/frontend/Window.java

Tárgymutató

- addMoreBombs
 - backend.Board, [10](#)
- addRecord
 - backend.Controller, [22](#)
 - backend.Data, [26](#)
- backend.BigBomb, [5](#)
 - BigBomb, [6](#)
 - changelcon, [7](#)
 - getValue, [7](#)
- backend.Board, [8](#)
 - addMoreBombs, [10](#)
 - Board, [9](#)
 - checkEnd, [10](#)
 - decHiddenTiles, [10](#)
 - findZerosAround, [10](#)
 - flagTile, [10](#)
 - generateBombs, [11](#)
 - getBombsAround, [11](#)
 - getBombsLeft, [11](#)
 - getBombsNr, [12](#)
 - getCols, [12](#)
 - getCorrespondingBomb, [12](#)
 - getFlagsLeft, [12](#)
 - getHiddenTiles, [12](#)
 - getRows, [12](#)
 - getXOffset, [13](#)
 - getYOffset, [13](#)
 - loadImages, [13](#)
 - paintComponent, [13](#)
 - resetFlags, [13](#)
 - resetGame, [13](#)
 - restart, [14](#)
 - revealEveryTile, [14](#)
 - revealTile, [14](#)
 - setBombsAroundNums, [14](#)
 - setBombsLeft, [14](#)
 - setController, [14](#)
 - setFlagsLeft, [15](#)
 - setStartPos, [15](#)
- backend.Bomb, [15](#)
 - Bomb, [16](#)
 - changelcon, [17](#)
 - execute, [17](#)
 - flag, [17](#)
 - getValue, [17](#)
 - reveal, [18](#)
 - revealEnd, [18](#)
 - setBombsAround, [18](#)
- backend.ClusterBomb, [19](#)
 - changelcon, [20](#)
 - ClusterBomb, [20](#)
 - execute, [20](#)
- backend.Controller, [21](#)
 - addRecord, [22](#)
 - clearRecords, [22](#)
 - closeGame, [22](#)
 - Controller, [21](#)
 - getData, [22](#)
 - getDiff, [23](#)
 - getGame, [23](#)
 - loadGame, [23](#)
 - loadRecords, [23](#)
 - pauseGame, [23](#)
 - resetWindowSize, [23](#)
 - saveGame, [24](#)
 - saveRecords, [24](#)
 - setDiff, [24](#)
 - setGame, [24](#)
 - setGameMenuBarEn, [24](#)
 - setMenuBar, [24](#)
 - setPanel, [24](#)
 - setSaveBtnEn, [25](#)
 - setWindow, [25](#)
 - setWindowSize, [25](#)
- backend.Data, [26](#)
 - addRecord, [26](#)
 - getColumnCount, [27](#)
 - getColumnName, [27](#)
 - getRowCount, [27](#)
 - getValueAt, [27](#)
- backend.Difficulty, [28](#)
 - bombs, [28](#)
 - cols, [28](#)
 - EASY, [29](#)
 - rows, [29](#)
 - setCustom, [29](#)
- backend.DifusedBomb, [30](#)
 - changelcon, [31](#)
 - DifusedBomb, [31](#)
 - reveal, [31](#)
- backend.Images, [38](#)
 - bigBomb, [39](#)
 - bomb, [39](#)
 - clusterBomb, [39](#)
 - difusedBomb, [39](#)
 - error, [39](#)
 - flag, [40](#)

- flag2, [40](#)
- hTile, [40](#)
- icon, [40](#)
- loadImages, [39](#)
- MineSweeperFont, [40](#)
- numbers, [40](#)
- redBomb, [40](#)
- resetBomb, [40](#)
- resetFlagBomb, [41](#)
- success, [41](#)
- timerFont, [41](#)
- unknown, [41](#)
- backend.Main, [41](#)
 - main, [42](#)
- backend.Record, [49](#)
 - getName, [50](#)
 - getTime, [50](#)
 - Record, [50](#)
- backend.ResetBomb, [53](#)
 - changelcon, [54](#)
 - execute, [54](#)
 - ResetBomb, [54](#)
- backend.ResetFlagBomb, [55](#)
 - changelcon, [56](#)
 - execute, [56](#)
 - ResetFlagBomb, [56](#)
- backend.Tile, [59](#)
 - bombsAround, [64](#)
 - changelcon, [61](#)
 - flag, [61](#)
 - getBombsAround, [61](#)
 - getCorrespondingIcon, [61](#)
 - getValue, [61](#)
 - getW, [62](#)
 - icon, [64](#)
 - isFlagged, [62](#), [64](#)
 - isRevealed, [64](#)
 - loadIcon, [62](#)
 - master, [64](#)
 - paintComponent, [62](#)
 - reveal, [62](#)
 - revealEnd, [63](#)
 - setBombsAround, [63](#)
 - setCoords, [63](#)
 - setFlagged, [63](#)
 - setRevealed, [64](#)
 - Tile, [60](#)
 - w, [65](#)
 - x, [65](#)
- backend.Time, [65](#)
 - compareTo, [66](#)
 - getM, [66](#)
 - getS, [67](#)
 - increase, [67](#)
 - It, [67](#)
 - Time, [66](#)
 - toString, [67](#)
- BigBomb
 - backend.BigBomb, [6](#)
- bigBomb
 - backend.Images, [39](#)
- Board
 - backend.Board, [9](#)
- Bomb
 - backend.Bomb, [16](#)
- bomb
 - backend.Images, [39](#)
- bombs
 - backend.Difficulty, [28](#)
- bombsAround
 - backend.Tile, [64](#)
- changelcon
 - backend.BigBomb, [7](#)
 - backend.Bomb, [17](#)
 - backend.ClusterBomb, [20](#)
 - backend.DifusedBomb, [31](#)
 - backend.ResetBomb, [54](#)
 - backend.ResetFlagBomb, [56](#)
 - backend.Tile, [61](#)
- checkEnd
 - backend.Board, [10](#)
- clearRecords
 - backend.Controller, [22](#)
- closeGame
 - backend.Controller, [22](#)
- ClusterBomb
 - backend.ClusterBomb, [20](#)
- clusterBomb
 - backend.Images, [39](#)
- cols
 - backend.Difficulty, [28](#)
- compareTo
 - backend.Time, [66](#)
- Controller
 - backend.Controller, [21](#)
- decHiddenTiles
 - backend.Board, [10](#)
- DifusedBomb
 - backend.DifusedBomb, [31](#)
- difusedBomb
 - backend.Images, [39](#)
- EASY
 - backend.Difficulty, [29](#)
- error
 - backend.Images, [39](#)
- execute
 - backend.Bomb, [17](#)
 - backend.ClusterBomb, [20](#)
 - backend.ResetBomb, [54](#)
 - backend.ResetFlagBomb, [56](#)
- FeedBackWindow
 - frontend.FeedBackWindow, [33](#)
- findZerosAround

- backend.Board, 10
- finished
 - frontend.Game, 35
- flag
 - backend.Bomb, 17
 - backend.Images, 40
 - backend.Tile, 61
- flag2
 - backend.Images, 40
- flagTile
 - backend.Board, 10
- frontend.FeedBackWindow, 32
 - FeedBackWindow, 33
- frontend.Game, 33
 - finished, 35
 - Game, 34
 - getBoard, 35
 - getTime, 35
 - init, 35
 - paintComponent, 35
 - reload, 35
 - restart, 36
 - running, 36
 - setFinished, 36
 - setRunning, 36
 - setStarted, 37
 - start, 37
 - started, 37
- frontend.MenuPanel, 42
 - initComponents, 43
 - MenuPanel, 43
 - setActionListeners, 43
- frontend.MyMenuBar, 44
 - initComponents, 45
 - MyMenuBar, 45
 - setActionListeners, 45
 - setGameBarEn, 45
 - setPauseBtnText, 45
 - setSaveBtnEn, 46
- frontend.MyMouseListener, 46
 - mouseReleased, 47
 - MyMouseListener, 47
 - setBoard, 47
- frontend.RecordPanel, 51
 - initComponents, 52
 - RecordPanel, 52
 - setActionListeners, 52
- frontend.SettingsPanel, 57
 - initComponents, 58
 - setActionListeners, 58
 - SettingsPanel, 57
 - submit, 58
- frontend.Timer, 68
 - getHeight, 69
 - getTime, 70
 - paintComponent, 70
 - run, 70
 - running, 70
 - setFinished, 70
 - setRunning, 71
 - start, 71
 - started, 71
 - Timer, 69
 - toString, 71
- frontend.VictoryPanel, 72
 - initComponents, 73
 - setActionListeners, 73
 - VictoryPanel, 72
- frontend.Window, 73
 - initComponents, 75
 - resetSize, 75
 - setPanel, 75
 - Window, 74
- Game
 - frontend.Game, 34
- generateBombs
 - backend.Board, 11
- getBoard
 - frontend.Game, 35
- getBombsAround
 - backend.Board, 11
 - backend.Tile, 61
- getBombsLeft
 - backend.Board, 11
- getBombsNr
 - backend.Board, 12
- getCols
 - backend.Board, 12
- getColumnCount
 - backend.Data, 27
- getColumnName
 - backend.Data, 27
- getCorrespondingBomb
 - backend.Board, 12
- getCorrespondingIcon
 - backend.Tile, 61
- getData
 - backend.Controller, 22
- getDiff
 - backend.Controller, 23
- getFlagsLeft
 - backend.Board, 12
- getGame
 - backend.Controller, 23
- getHeight
 - frontend.Timer, 69
- getHiddenTiles
 - backend.Board, 12
- getM
 - backend.Time, 66
- getName
 - backend.Record, 50
- getRowCount
 - backend.Data, 27
- getRows
 - backend.Board, 12

- getS
 - backend.Time, 67
- getTime
 - backend.Record, 50
 - frontend.Game, 35
 - frontend.Timer, 70
- getValue
 - backend.BigBomb, 7
 - backend.Bomb, 17
 - backend.Tile, 61
- getValueAt
 - backend.Data, 27
- getW
 - backend.Tile, 62
- getXOffset
 - backend.Board, 13
- getYOffset
 - backend.Board, 13
- hTile
 - backend.Images, 40
- icon
 - backend.Images, 40
 - backend.Tile, 64
- increase
 - backend.Time, 67
- init
 - frontend.Game, 35
- initComponents
 - frontend.MenuPanel, 43
 - frontend.MyMenuBar, 45
 - frontend.RecordPanel, 52
 - frontend.SettingsPanel, 58
 - frontend.VictoryPanel, 73
 - frontend.Window, 75
- isFlagged
 - backend.Tile, 62, 64
- isRevealed
 - backend.Tile, 64
- loadGame
 - backend.Controller, 23
- loadIcon
 - backend.Tile, 62
- loadImages
 - backend.Board, 13
 - backend.Images, 39
- loadRecords
 - backend.Controller, 23
- It
 - backend.Time, 67
- main
 - backend.Main, 42
- master
 - backend.Tile, 64
- MenuPanel
 - frontend.MenuPanel, 43
- MineSweeperFont
 - backend.Images, 40
- mouseReleased
 - frontend.MyMouseListener, 47
- MyMenuBar
 - frontend.MyMenuBar, 45
- MouseListener
 - frontend.MyMouseListener, 47
- numbers
 - backend.Images, 40
- paintComponent
 - backend.Board, 13
 - backend.Tile, 62
 - frontend.Game, 35
 - frontend.Timer, 70
- pauseGame
 - backend.Controller, 23
- Record
 - backend.Record, 50
- RecordPanel
 - frontend.RecordPanel, 52
- redBomb
 - backend.Images, 40
- reload
 - frontend.Game, 35
- ResetBomb
 - backend.ResetBomb, 54
- resetBomb
 - backend.Images, 40
- ResetFlagBomb
 - backend.ResetFlagBomb, 56
- resetFlagBomb
 - backend.Images, 41
- resetFlags
 - backend.Board, 13
- resetGame
 - backend.Board, 13
- resetSize
 - frontend.Window, 75
- resetWindowSize
 - backend.Controller, 23
- restart
 - backend.Board, 14
 - frontend.Game, 36
- reveal
 - backend.Bomb, 18
 - backend.DifusedBomb, 31
 - backend.Tile, 62
- revealEnd
 - backend.Bomb, 18
 - backend.Tile, 63
- revealEveryTile
 - backend.Board, 14
- revealTile
 - backend.Board, 14
- rows

- backend.Difficulty, 29
- run
 - frontend.Timer, 70
- running
 - frontend.Game, 36
 - frontend.Timer, 70
- saveGame
 - backend.Controller, 24
- saveRecords
 - backend.Controller, 24
- setActionListeners
 - frontend.MenuPanel, 43
 - frontend.MyMenuBar, 45
 - frontend.RecordPanel, 52
 - frontend.SettingsPanel, 58
 - frontend.VictoryPanel, 73
- setBoard
 - frontend.MyMouseListener, 47
- setBombsAround
 - backend.Bomb, 18
 - backend.Tile, 63
- setBombsAroundNums
 - backend.Board, 14
- setBombsLeft
 - backend.Board, 14
- setController
 - backend.Board, 14
- setCoords
 - backend.Tile, 63
- setCustom
 - backend.Difficulty, 29
- setDiff
 - backend.Controller, 24
- setFinished
 - frontend.Game, 36
 - frontend.Timer, 70
- setFlagged
 - backend.Tile, 63
- setFlagsLeft
 - backend.Board, 15
- setGame
 - backend.Controller, 24
- setGameBarEn
 - frontend.MyMenuBar, 45
- setGameMenuBarEn
 - backend.Controller, 24
- setMenuBar
 - backend.Controller, 24
- setPanel
 - backend.Controller, 24
 - frontend.Window, 75
- setPauseBtnText
 - frontend.MyMenuBar, 45
- setRevealed
 - backend.Tile, 64
- setRunning
 - frontend.Game, 36
 - frontend.Timer, 71
- setSaveBtnEn
 - backend.Controller, 25
 - frontend.MyMenuBar, 46
- setStarted
 - frontend.Game, 37
- setStartPos
 - backend.Board, 15
- SettingsPanel
 - frontend.SettingsPanel, 57
- setWindow
 - backend.Controller, 25
- setWindowSize
 - backend.Controller, 25
- start
 - frontend.Game, 37
 - frontend.Timer, 71
- started
 - frontend.Game, 37
 - frontend.Timer, 71
- submit
 - frontend.SettingsPanel, 58
- success
 - backend.Images, 41
- Tile
 - backend.Tile, 60
- Time
 - backend.Time, 66
- Timer
 - frontend.Timer, 69
- timerFont
 - backend.Images, 41
- toString
 - backend.Time, 67
 - frontend.Timer, 71
- unknown
 - backend.Images, 41
- VictoryPanel
 - frontend.VictoryPanel, 72
- w
 - backend.Tile, 65
- Window
 - frontend.Window, 74
- x
 - backend.Tile, 65