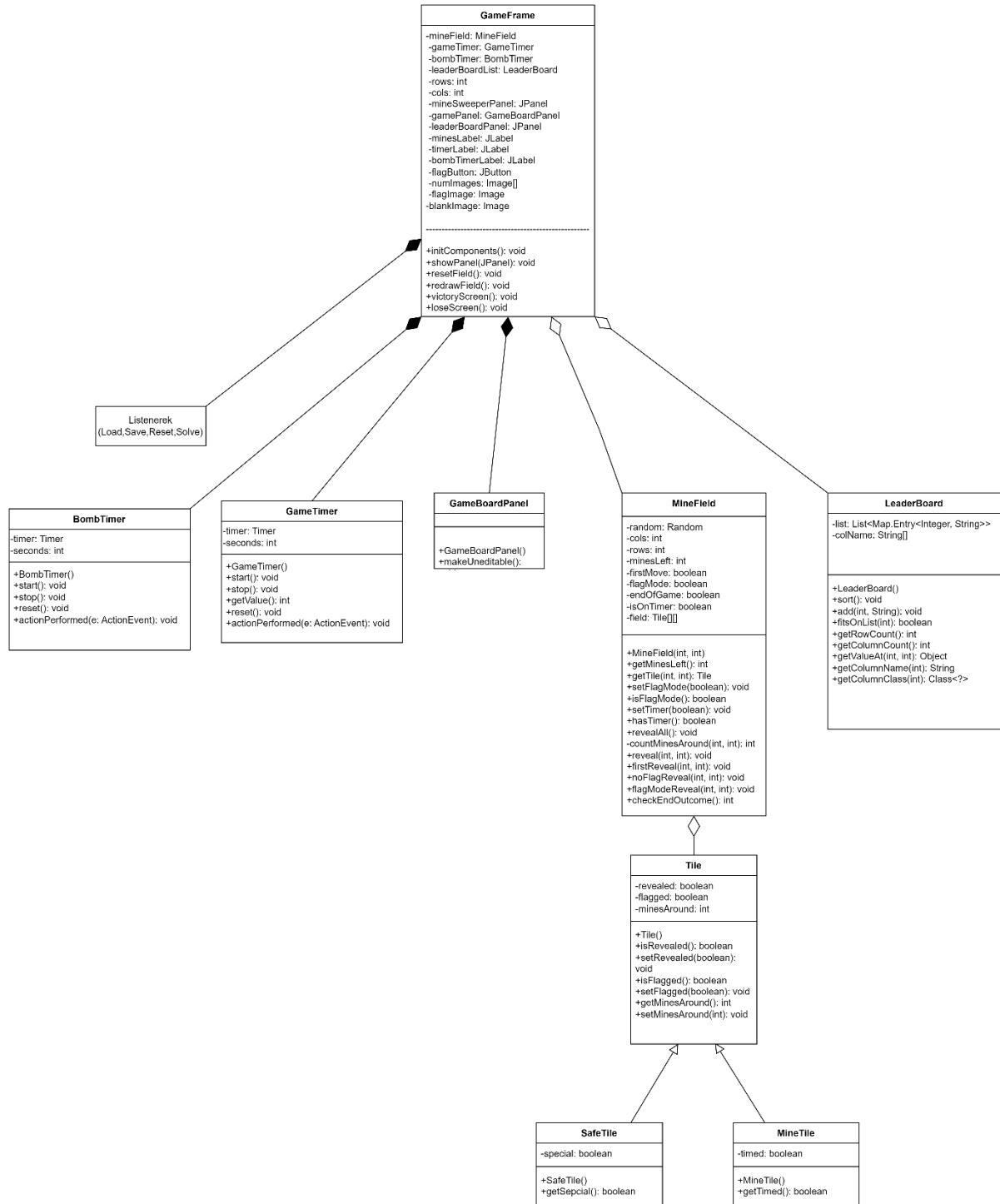


QRRHP7 – Aknakereső dokumentáció

Uml



Osztályleírások

----- Tile (Mező): -----

Az osztály az aknamezőt képező alapegységeknek (bomba és üres mező) alapjául szolgáló absztrakt osztály. Az osztály implementálja a Serializable interfészt, így a leszármazottak példányai szerializálhatóak lesznek.

Mezők:

- boolean revealed:
jelzi, hogy a mező fel lett-e fedve
- boolean flagged:
jelzi, hogy a mező meg lett-e jelölve zászlóval

Metódusok:

- boolean isRevealed():
Ellenőrzi, hogy a csempe fel lett-e fedve
Visszatérési érték: true, ha a mező felfedett, különben false
- void setRevealed(boolean revealed):
Beállítja a csempe felfedettséget
Paraméter: revealed - true, ha a mező fel van fedve, különben false
- boolean isFlagged():
Ellenőrzi, hogy a mező meg lett-e jelölve zászlóval
Visszatérési érték: true, ha a mező meg van jelölve zászlóval, különben false
- void setFlagged(boolean flagged):
Beállítja a csempe megjelöltségét
Paraméter: flagged - true, ha a mező meg van jelölve, különben false
- abstract int getMinesAround():
Abstract metódus. Megvalósítása megadja a mező körüli aknák számát
Visszatérési érték: -1, ha a mező akna, különben 0-8
- abstract void setMinesAround(int minesAround):
Abstract metódus. Megvalósítása beállítja a mező körüli aknák számát
Paraméter: minesAround - A mező körüli aknák száma (0-8). -1 érték esetén az akna jelző

MineTile (Aknamező):

A MineTile osztály egy speciális Tile, amely lehet akna, és esetleg időzített akna is.

Mezők:

- boolean isTimed:
Jelzi, hogy az akna időzített-e vagy sem

Metódusok:

- public MineTile():
Konstruktor
A konstruktor véletlenszerűen eldönti, hogy az akna időzített-e
Alapvetően 10% eséllyel időzített akna lesz
- int getMinesAround():
Megadja a mező körüli aknák számát
Visszatérési érték: Mindig -1, mert ez egy akna
- boolean getTimed():
Ellenőrzi, hogy az akna időzített-e
Visszatérési érték: true, ha az akna időzített, különben false
- void setMinesAround(int m):
Az absztrakt őssztály miatt szükséges, de ebben az osztályban nem csinál semmit
Minden esetben -1 értéket ad meg, mivel ez egy akna

SafeTile (Biztonságos mező):

A SafeTile osztály egy speciális Tile, amely nem bomba, és emellett lehet sepciális hatása is: pár zászlóval megjelölt mezőről eltünteti a jelölést.

Mezők:

- int minesAround:
Megadja a mező körüli aknák számát.
- boolean special:
Jelzi, hogy a mező rendelkezik-e speciális tulajdonsággal.

Metódusok:

- public SafeTile():
Konstruktor
A konstruktor véletlenszerűen eldönti, hogy a mező rendelkezik-e speciális tulajdonsággal.
Alapvetően 3% eséllyel rendelkezik speciális tulajdonsággal.
- int getMinesAround():
Megadja a mező körüli aknák számát.
Visszatérési érték: A mező körüli aknák száma.
- void setMinesAround(int m):
Beállítja a mező körüli aknák számát.
Paraméter: m - A mező körüli aknák száma.
- boolean getSpecial():
Ellenőrzi, hogy a mező rendelkezik-e speciális tulajdonsággal.
Visszatérési érték: true, ha a mező rendelkezik speciális tulajdonsággal, különben false.

----- **LeaderBoard (Toplista):** -----

A LeaderBoard osztály egy táblázatot reprezentál, amely tárolja és kezeli a legjobb eredményeket. A játék bezárásakor automatikusan elmentve, indulásakor automatikusan beolvasva.

Mezők:

- List<Map.Entry<Integer, String>> list:
Egy lista, amely tartalmazza a legjobb eredményeket, ahol a kulcs az idő, az érték pedig a név. (Azért így, mert a lista rendezése gyorsabb kulcsok alapján, mint értékek alapján.)
- String[] colName:
A táblázat oszlopainak neveit tartalmazó tömb.

Metódusok:

- public LeaderBoard():
Konstruktor
Inicializálja a listát, amely tárolja a legjobb eredményeket.
- public void sort():
A listát növekvő sorrendbe rendezi az idő alapján.
Értesíti a táblázat modellt, hogy adatai megváltoztak.
- public void add(int score, String name):
Hozzáad egy új eredményt a listához, és szükség esetén frissíti és rendezi is.

Paraméterek: score - Az elért idő, name - A játékos neve.

- public boolean fitsOnList(int score):

Ellenőrzi, hogy az új eredmény felkerülhet-e a listára.

Visszatérési érték: true, ha az új eredmény felkerülhet, különben false.

Metódusok (AbstractTableModel felülírása):

- public int getRowCount():

Visszaadja a sorok számát a táblázatban.

Visszatérési érték: A sorok száma a listában.

- public int getColumnCount():

Visszaadja az oszlopok számát a táblázatban.

Visszatérési érték: 2, az oszlopok száma.

- public Object getValueAt(int rowIndex, int columnIndex):

Visszaadja a táblázatban található értéket adott sorban és oszlopban.

Paraméterek: rowIndex - A sor indexe, columnIndex - Az oszlop indexe.

Visszatérési érték: Az érték a megadott sorban és oszlopban.

- public String getColumnName(int index):

Visszaadja az oszlop nevét az adott indexen.

Paraméter: index - Az oszlop indexe.

Visszatérési érték: Az oszlop neve.

- public Class<?> getColumnClass(int col):

Visszaadja az oszlop típusát az adott indexen.

Paraméter: col - Az oszlop indexe.

Visszatérési érték: Az oszlop típusa.

MineField (Aknakereső):

A MineField osztály egy aknakereső játéket reprezentál, amelyet Tile mezők alkotnak. Itt hajtódik végre a játék logikája.

Mezők:

- Random random:

Többször felhasznált véletlenszám generátor.

- int cols:

Az oszlopok száma a játékban.

- int rows:

A sorok száma a játékban.

- int minesLeft:

A hátralevő aknák száma a játékban.

- boolean firstMove:

Igaz, ha minden mező be van fedve az első kattintás előtt.

- boolean flagMode:

Igaz, ha a játék zászló módban van.

- boolean endOfGame:

Igaz, ha a játék véget ért.

- boolean isOnTimer:

Igaz, ha egy időzített aknát kattintottak meg, és még nincs hatástalanítva.

- Tile[][] field:

A játékteret reprezentáló mezők tömbje.

Metódusok:

- public MineField(int r, int c):

Konstruktor

Inicializálja a játékteret biztonságos mezőkkel.

- public int getMinesLeft():

Visszaadja a hátralevő aknák számát a játékban.

Visszatérési érték: A hátralevő aknák száma.

- public Tile getTile(int r, int c):

Visszaadja a megadott sorban és oszlopban található mezőt a játéktérben.

Paraméterek: r - sor index, c - oszlop index.

Visszatérési érték: A megadott mező a játéktérben.

- public void setFlagMode(boolean b):

Beállítja a zászló módot a megadott értékre.

Paraméter: b - True, ha zászló mód bekapcsolva, különben false.

- public boolean isFlagMode():

Ellenőrzi, hogy a játék zászló módban van-e.

Visszatérési érték: True, ha zászló mód bekapcsolva, különben false.

- public void setTimer(boolean b):

Beállítja az időzítő állapotát a megadott értékre.

Paraméter: b - True, ha az időzítő bekapcsolva, különben false.

- public boolean hasTimer():

Ellenőrzi, hogy az időzítő be van-e kapcsolva.

Visszatérési érték: True, ha az időzítő be van kapcsolva, különben false.

- public void revealAll():

Az összes mezőt felfedett állapotba állítja.

- private int countMinesAround(int row, int col):

Megszámolja a biztonságos mező körüli aknákat.

Paraméterek: row - A mező sor indexe, col - A mező oszlop indexe.

Visszatérési érték: A környező aknák száma.

Metódusok (Felfedezés funkciók):

- public void reveal(int row, int col):

Felfed egy mezőt a játéktérben.

Paraméterek: row - A mező sor indexe, col - A mező oszlop indexe.

- public void firstReveal(int row, int col):

Az első felfedést kezeli, inicializálja a játéktér. Itt generálódnak a bombák.

Paraméterek: row - A mező sor indexe, col - A mező oszlop indexe.

- public void noFlagReveal(int row, int col):

A nem zászló módban történő felfedezést kezeli.

Paraméterek: row - A mező sor indexe, col - A mező oszlop indexe.

- public void flagModeReveal(int row, int col):

A zászló módban történő felfedezést kezeli.

Paraméterek: row - A mező sor indexe, col - A mező oszlop indexe.

Metódusok (Végső eredmény ellenőrzése):

- public int checkEndOutcome():

Ellenőrzi a játék végső eredményét.

Visszatérési érték: -1, ha a játékos veszített, 0, ha a játék még nem ért véget, 1, ha a játékos nyert.

GameFrame (Főablak):

A GameFrame osztály egy Java Swing alkalmazás, ami a Főablakot reprezentálja egy Aknakereső játékban. Ez a JFrame osztályból származik, biztosítva a játék grafikus felhasználói felületét. Az osztály tartalmaz olyan komponenseket, mint egy aknamező, időzítő, toplista és különböző UI elemek.

Mezők:

-mineField:

Az aknamezőt reprezentálja a játékban.

-gameTimer:

A játék időzítője.

-bombTimer:

Az időzített bombákhoz tartozó időzítő.

-leaderBoardList:

Toplista, tárolja a legjobb eredményeket és játékosneveket.

-rows:

A mező sorainak száma.

-cols: A

mező oszlopainak száma.

-mineSweeperPanel:

A fő Aknakereső játéktérületet tartalmazó panel. (gamePanel + statusBar)

-gamePanel:

A játéktáblát reprezentáló panel.

-leaderBoardPanel:

A toplistát megjelenítő panel.

-minesLabel:

JLabel, amely az jelöletlen aknák számát jelzi.

-timerLabel:

Statikus JLabel, amely megjeleníti a játék időzítőjét.

-bombTimerLabel:

Statikus JLabel, amely megjeleníti az időzített bombák időzítőjét.

-flagButton:

Gomb, amellyel zászló módba és abból vissza válthatunk.

-numImages:

Képek tömbje, amelyek a felfedett cellák körüli aknák számát reprezentálják (0-8).

-flagImage:

A zászlót ábrázoló kép.

-blankImage:

A felfedetlen cellát ábrázoló kép.

Metódusok:

-public void initComponents():

Inicializálja a játék ablak összes komponensét, beleértve az aknamezőt, az időzítőket, a felhasználói felületi elemeket és a menüsört.

-public void showPanel(JPanel panel):

Átvált a megadott panelre.

Paraméter: A megjelenítendő panel.

-public void resetField():

Visszaállítja a játékkeret, újraindítja az időzítőket, és generál egy új aknamezőt.

-public void redrawField():

Frissíti a játéktábla megjelenését az aknamező jelenlegi állapotától függően.

-public void victoryScreen():

Megjeleníti a győzelem üzenetet, leállítja az időzítőket, és lehetővé teszi a játékosnak, hogy megadja a nevét a toplistához.

-loseScreen():

Megjeleníti a vereség üzenetet, és leállítja az időzítőket, amikor a játék véget ér.

Belső osztályok (A GameFrame változóinak elérésének érdekében):

GameBoardPanel:

A GameBoardPanel osztály a Minesweeper alkalmazásban található játéktáblát reprezentálja. Ez az osztály a GameFrame osztályon belül található, a JPanel osztály leszármazottja. A játéktábla megjelenítéséért felel, gombokat hoz létre, amelyek az egyes mezőket képviselik.

-public GameBoardPanel():

Konstruktor. Beállítja a elrendezést és létrehozza a gombokat minden egyes mezőhöz egy GridLayoutban.

-public void makeUneditable():

Ez a metódus eltávolít minden ActionListener-t a gombokról a játéktáblán belül, így a gombok nem lesznek megnyomhatók. Ez akkor hasznos, amikor a játék véget ér, és további interakciók nem engedélyezettek.

FlagListener:

Ez a belső osztály implementálja az ActionListener-t a zászlóval kapcsolatos műveletek kezeléséhez.

-public void actionPerformed(ActionEvent ae):

Ez a metódus akkor fut le, amikor a zászló gombra kattintanak. Frissíti a zászlógomb ikonját a frontend-en, és beállítja a zászló mód állapotát a játéktérben (mineField).

CellListener:

Ez a belső osztály implementálja az ActionListener-t a gombra kattintások kezeléséhez az egyes cellákban. Felfedi a kattintott cella tartalmát, elindítja vagy leállítja az időzítőket a játék állapotától függően, frissíti a megjelenítést, és ellenőrzi a játék végét.

-public void actionPerformed(ActionEvent ae):

Ez a metódus akkor fut le, amikor egy cella gombra kattintanak. Felfedi a cella tartalmát, elindítja vagy leállítja az időzítőket, frissíti a megjelenítést, és ellenőrzi a játék végét.

GameTimer:

A GameTimer osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. Egy időzítőt valósít meg, amely nyomon követi az eltelt időt a játék során.

Metódusok:

-public GameTimer():

Az osztály konstruktora, inicializálja a seconds változót és létrehoz egy időzítőt egy másodperces késleltetéssel.

-public void start():

Elindítja az időzítőt, ha az létezik.

-public void stop():

Leállítja az időzítőt.

-public int getValue():

Visszaadja az eltelt időt másodpercekben.

-public void reset():

Visszaállítja az eltelt időt nullára és frissíti a megjelenítést.

-public void actionPerformed(ActionEvent e):

A Timer eseménykezelőjeként működik, növeli az eltelt időt másodpercenként és frissíti a megjelenítést.

Serializáció:

-private void writeObject(ObjectOutputStream out):

Serializációs metódus, menti az eltelt időt.

-private void readObject(ObjectInputStream in): Deserializációs metódus, visszaállítja az eltelt időt és létrehozza az új időzítőt.

BombTimer:

A BombTimer osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. Ez egy időzítő, amely egy visszaszámlálást valósít meg a játékban lévő időzített bombához.

Metódusok:

-public BombTimer():

Az osztály konstruktora, inicializálja a seconds változót, elrejtja a bomba időzítőjelzőt, és létrehoz egy időzítőt egy másodperces késleltetéssel.

-public void start():

Elindítja az időzítőt, megjelenítve a bomba JLabelt.

-public void stop():

Leállítja az időzítőt és elrejtja a bomba JLabelt.

-public void reset():

Visszaállítja a bomba JLabelt és az időzítőt az eredeti értékre.

-public void actionPerformed(ActionEvent e):

A Timer eseménykezelőjeként működik, csökkenti a hátralévő időt másodpercenként, frissíti a bomba JLabelt, és végrehajtja a veszteskor teendő dolgokat, ha az idő lejár.

Serializáció:

-private void writeObject(ObjectOutputStream out):

Serializációs metódus, menti a hátralévő időt.

-private void readObject(ObjectInputStream in):

Deserializációs metódus, visszaállítja a hátralévő időt és létrehozza az új időzítőt.

ResetListener:

A ResetListener osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. A gomb lenyomására a játékkeret visszaállítja a kezdeti állapotba.

-public void actionPerformed(ActionEvent ae):

Az osztály egyetlen metódusa, amely akkor hívódik meg, amikor a reset gombot lenyomják.

Ekkor meghívja a resetField metódust, ami visszaállítja a játékkeret.

SolveListener:

A SolveListener osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. A gomb lenyomására az összes mezőt felfedve megoldja a játékot.

-public void actionPerformed(ActionEvent ae):

Az osztály egyetlen metódusa, amely akkor hívódik meg, amikor a solve gombot lenyomják. Ekkor meghívja a mineField.revealAll() metódust, majd leállítja a játékidőt, újrarajzolja a játékkeret és újrarajzolja az ablakot.

SaveListener:

A SaveListener osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. A gomb lenyomására lehetőséget ad a játék állapotának elmentésére fájlba.

- public void actionPerformed(ActionEvent ae):

Az osztály egyetlen metódusa, amely akkor hívódik meg, amikor a save gombot lenyomják. Ekkor egy JFileChooser segítségével lehetőség nyílik a felhasználónak fájl választani, majd elmenti a játékállapotot a kiválasztott fájlba.

LoadListener:

A LoadListener osztály egy belső osztály a Minesweeper alkalmazásban, mely a GameFrame osztályon belül található. A gomb lenyomására lehetőséget ad a korábban elmentett játékállapot betöltésére.

- public void actionPerformed(ActionEvent ae):

Az osztály egyetlen metódusa, amely akkor hívódik meg, amikor a load gombot lenyomják. Ekkor egy JFileChooser segítségével lehetőség nyílik a felhasználónak fájl választani, majd betölti a játékállapotot a kiválasztott fájlból.

Felhasználói kézikönyv

A játék indulásakor automatikusan nekiláthatunk a játéknak. A kezdéshez kattintsunk egy tetszőleges mezőre. Első kattintásra garantáltan nagyobb mezőcsoportot fedünk fel, nem kell tartani a bombára kattintástól.

Ezek után a célunk a bombák zászlóval való megjelölése és a nem bomba mezők felfedése lesz. Ezt hátráltatják a különleges nem bomba mezők, amelyek kattintásra eltüntetnek három zászlójelölést, így érdemes figyelni ezekre. Ezt ellensúlyozzák az olyan időzített bombák, amelyeket, ha felfedjük nem veszítjük el egyből a játékot. Ilyenkor van 5 másodpercünk egy bombát zászlóval megjelölni, különben a játék véget ér.

A navigáció menüsor segítségével történik. A File menüpontra majd a Save almenüpontra kattintva, majd egy létező .txt fájlt kiválasztva elmenthetjük a jelenlegi játékállapotot. Ezután kiléphetünk, indíthatunk új játékot is, majd ugyanitt a Load almenüpontra kattintva egy elmentett játékot tartalmazó .txt fájlt kiválasztva visszatölthetjük azt. Ekkor onnan folytathatjuk, ahol abbahagytuk.

A Game menüpont szintén két almenüt tartalmaz. A Reset almenüre kattintva egy új játékot kezdhünk teljesen új aknamezővel. A Solve almenüvel pedig megoldhatjuk (és megjeleníthetjük) a jelenlegi játék megoldását. Természetesen ezután már nem oldhatjuk meg a jelenlegi mezőt.

Utolsó menüpont a LeaderBoard, amelynek LeaderBoard almenüpontjával megjeleníthetjük a toplistát, a Minesweeper almenüvel pedig magát a játékot.

A játék elvesztésekor értesít minket a játék és már nem tudunk játékmezőre kattintani. Ekkor a Reset-el kezdhethünk új játékot. Ha nyerünk a játék akkor is értesít minket. Lehetőséget kapunk nevet megadni, ami felkerül a toplistára, ha elég gyorsak voltunk. Ha nem adunk meg nevet az eredmény semmiképpen nem kerül fel.

Sok sikert a játékhoz!