Első beadandó feladat

Készítette

Hujber Ferenc Kristóf

BWSOU0@INF.ELTE.HU

Feladat:

Aknamező

Készítsünk programot a következő játékra.

A játékban egy tengeralattjárót kell irányítanunk a képernyőn (balra, jobbra, fel, illetve le), amely felett ellenséges hajók köröznek, és folyamatosan aknákat dobnak a tengerbe. Az aknáknak három típusa van (könnyű, közepes, nehéz), amely meghatározza, hogy milyen gyorsan süllyednek a vízben (minél nehezebb, annál gyorsabban).

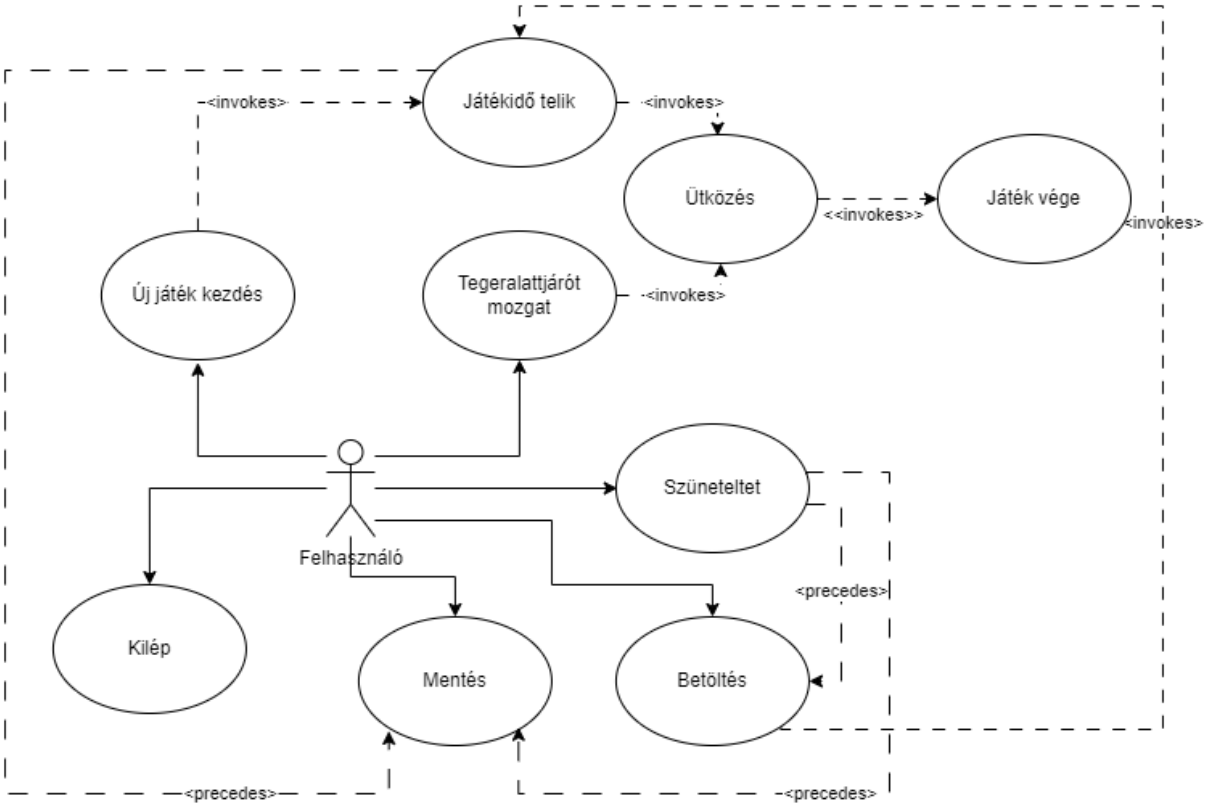
Az aknákat véletlenszerűen dobják a tengerbe, ám mivel a hajóskapitányok egyre türelmetlenebbek, egyre gyorsabban kerül egyre több akna a vízbe. A játékos célja az, hogy minél tovább elkerülje az aknákat. A játék addig tart, ameddig a tengeralattjárót el nem találta egy akna.

A program biztosítson lehetőséget új játék kezdésére, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog semmi a játékban). Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

Elemzés:

- Megvalósítás egyablakos Windows Forms applikációként, grafikus felülettel. A játékteret egy rácstáblázattal jelenítjük meg.
- A játék indításakor a játékos új játékot indíthat
- A felhasználó tudja a játékot szüneteltetni. Ekkor a következő menüpontokat veheti igénybe: mentés, betöltés, új játék indítása. A játék futása közben nincs szükség a menü elérhetőségére.
    - A mentést és a betöltést dialógusablakkal valósítjuk meg.
- A felhasználó tudja mozgatni a tengeralattjárót (azaz a játékos karaktert) a játéktéren.

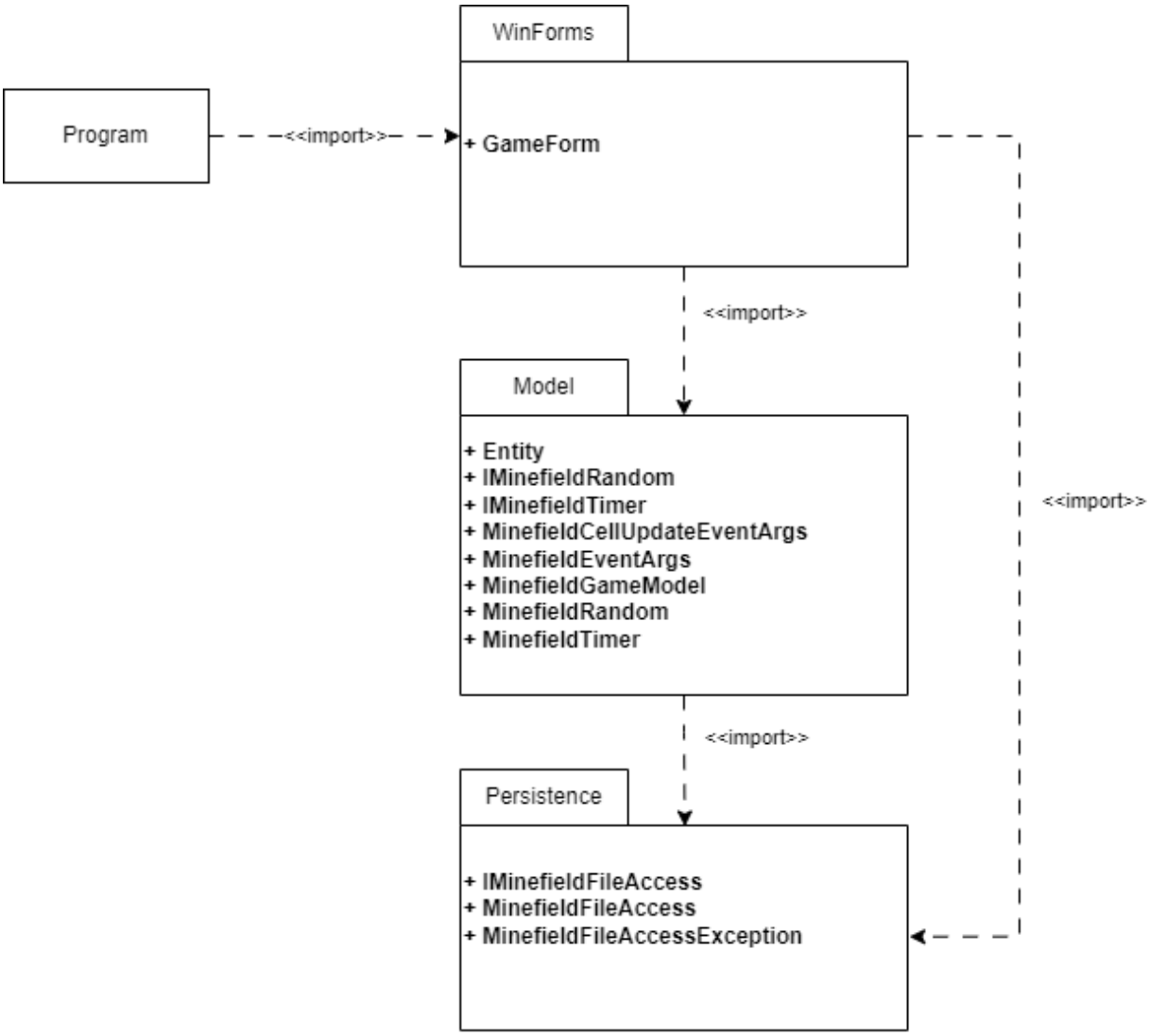
- A játék véget érhet a következő esetekben:
  - A tengeralattjáró nekimegy egy aknának.
  - Az idő teltével egy akna a tengeralattjáróra esik.
- A használati esetekért lásd az 1. ábrát.



1. ábra: Használai eset diagram

Tervezés:

- Programszerkezet: háromrétegű architektúra:
  - Felületfüggetlen:
    - **modell**: Minefield.Model névtér
    - **perzisztencia**: Minefield.Persistence névtér
  - Windows Forms Application:
    - **nézet**: Minefield.WinForms névtér
- A csomagdiarammért lásd a 2. ábrát.

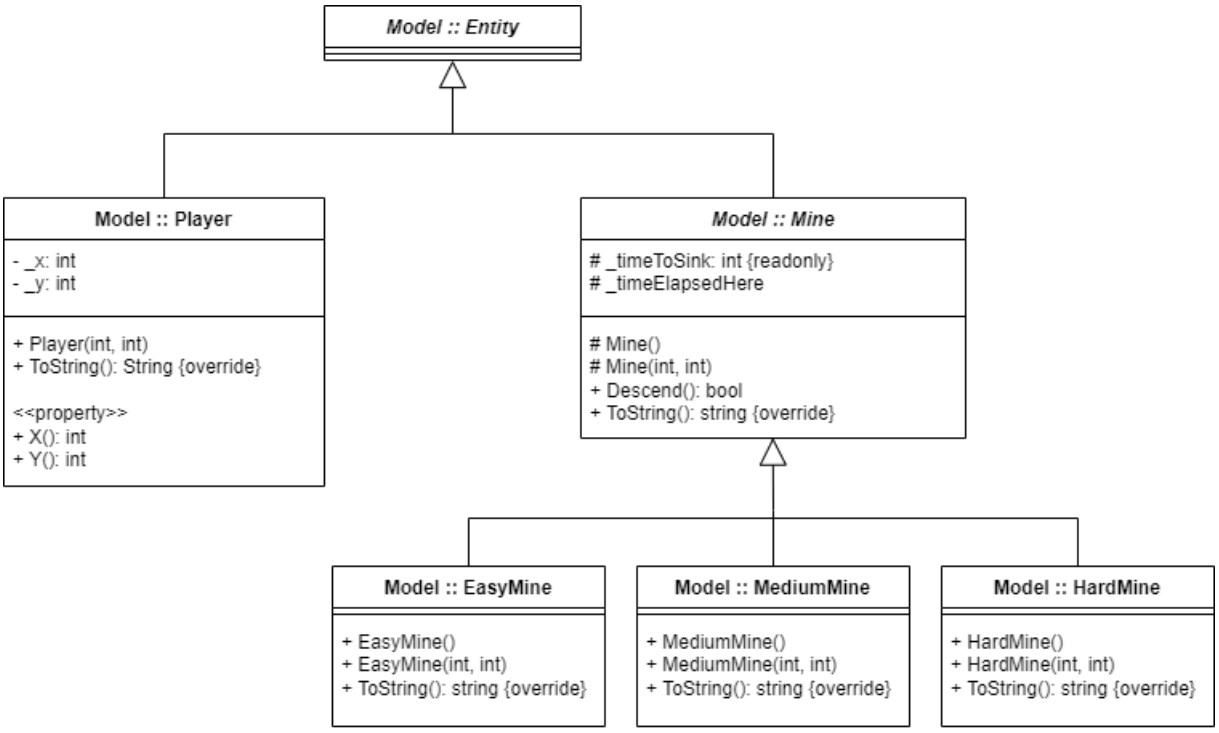


2. ábra: Csomagdiagramm

Modell

- Az üzleti logikát és a játéktér reprezentációját a MinefieldGameModel osztály tartalmazza.
- Az üzleti logika legfontosabb elemei:
  - Játék indítása, szüneteltetése, leállítása, mentés/betöltés kezelése
  - A játéktér változásainak kezelése, mint a játékos karakter mozgatása, aknák létrehozása és mozgatása, ütközés ellenőrzése
  - Időzítő eseményének kezelése
- A játékteret egy Entitásmátrix (_entities) és a játékidő (_gameTime) reprezentálja. Ezen felül tárolunk a játékos entitásra egy referenciát (_player) annak gyors eléréshez a mozgatás esetén.

- Az entitás típus és leszármazottai:
  - A játéktér résztvevőinek absztrakt ősosztálya az Entity. Ennek Leszármazottja a Player, azaz a játékos karakter típusa és a Mine, azaz az akna típusa.
  - A Mine absztrakt szülőosztály. Ennek alosztályai a ténylegesen példányosított EasyMine, MediumMine, HardMine.
  - Az Entity típusok osztálydiagramjaiért lásd a 3. ábrát.
- A modell események által kommunikál a nézet felé.
  - TimeAdvanced: A futó játékban a játékidő haladásáról tájékoztat.
  - GameOver: A játék végéről tájékoztat, ami ütközéskor keletkezik be.
  - CellUpdate: A játéktér egy cellájának változásáról tájékoztat.
- Esemény-argumentum típusok:
  - MinefieldEventArgs: Tartalmazza a játékidőt.
  - MinefieldCellUpdateEventArgs: Tartalmazza az adott mező koordinátáit.
- MinefieldGameModel implementálja az IDisposable iterfészt, mert a _gameTimer field-je Disposable.



3. ábra: Entity osztály és leszármazottainak osztálydiagrammja

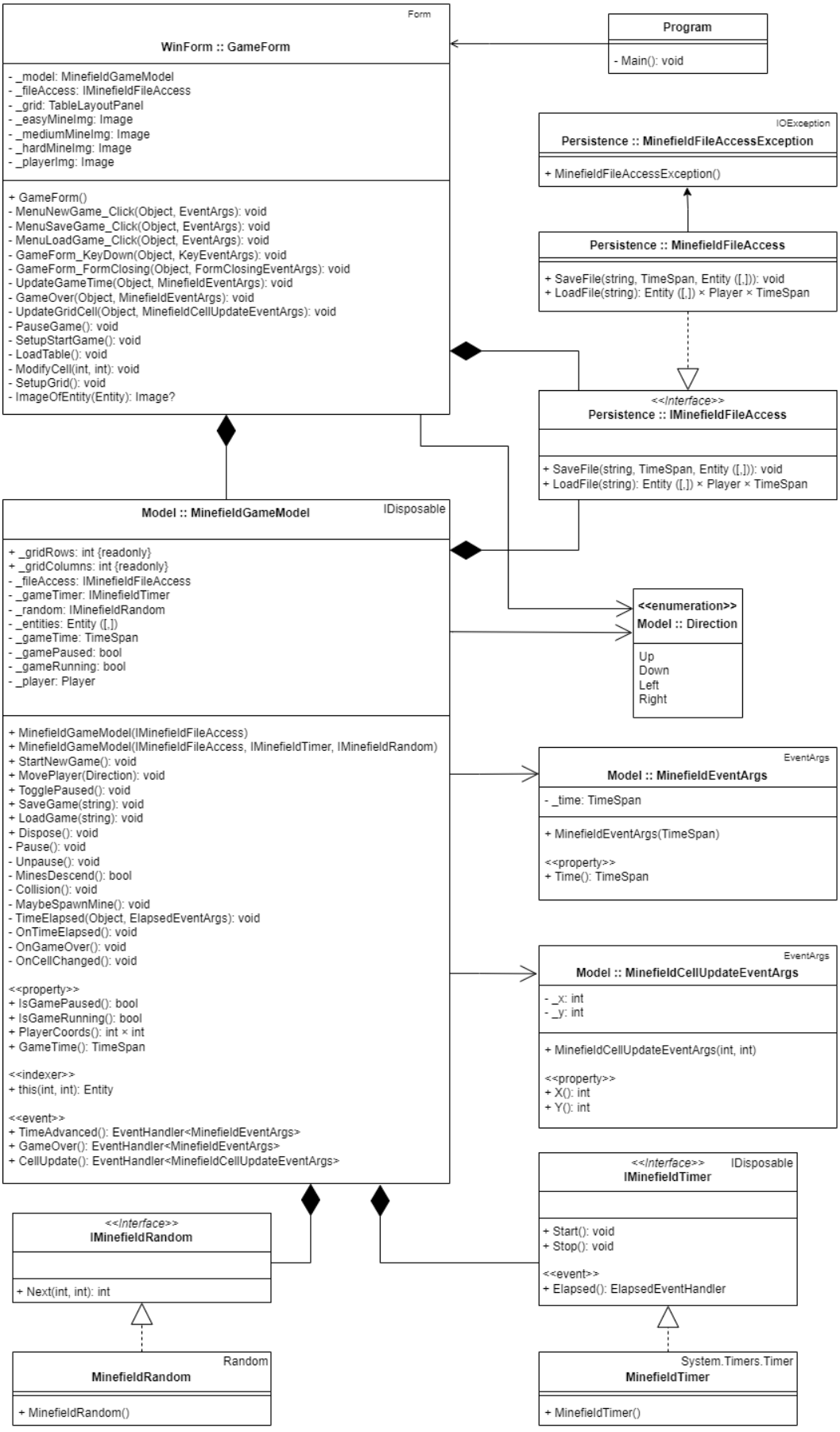
Perzisztencia

- Feladata a játékállás mentésének betöltése, valamint egy létező mentés-fájlból a játékállás beolvasása.
- A mentés és betöltés metódusait az IMinefieldFileAccess interfész adja meg:
  - SaveFile
  - LoadFile
- Az utóbbi fájlkezelő metódusok megvalósítását a MinefieldFileAccess tartalmazza.
- A fájlkezelés közben fellépő hiba esetén MinefieldFileAccessException kivétel váltódik ki, ami az IOException leszármazottja.
- Egy mentés-fájl (.MFS mint MineFieldSave) szerkezete a következő soronként:
  - játékidő milliszekundumban
  - játéktér sorszáma
  - játéktér oszlopszáma
  - a játéktér entitás-mátrixának elemei sorfolytonosan (1 entitás / 1 sor a fájlban) a következő szerint
    - 0 ha null
    - az entitás string-reprezentációja ha nem null
      - lásd: felüldefiniált ToString() metódusok

Nézet

- A nézetet a GameForm osztály valósítja meg.
- A nézet aggregálja a modell és a perzisztencia egy-egy példányát. Függőségi befecskendezést végez: a példányosított IMinefieldFileAccess típusú _fileAccess-t adja át a modell konstruktorának az utóbbi példányosításánál.
- A játéktér megjelenítésének alapja a TableLayoutPanel típusú rács (_grid), aminek a celláiba illesztünk PictureBox objektumokat az entitások reprezentálására. Az ablak további lényeges elemei a játékidőt mutató _statusStrip, és a játék szünetelésekor megjelenő _pauseMenuStrip.
- A nézet eseménykezelői:
  - Kezelik a felhasználó által kiváltott eseményeket, ideértve az irányító billentyűk és a menüpontok használatát.
  - Kezelik a modell eseményeit, tehát megjeleníti a játéktér változásai.

Az átfogó osztálydiagrammért lásd a 4. ábrát.



4. ábra: Osztálydiagramm

Az Entity osztály és leszármazottai nem szerepelnek ezen az ábrán az áttekinthetőségért. Lásd a 3. ábrát. Belátható, hogy a modell az Entity-vel és a Player-rel kompozícióban, a többivel asszociációban áll. Ezen felül a GameForm, MinefieldFileAccess, IMinefieldFileAccess, és az esemény-argumentum típusok is asszociációban állnak az Entity-típusokkal.

A modell 3-paraméteres konstruktora lehetőséget ad arra, hogy függőségi befecskendezéssel adjuk meg az időzítőt és a véletlenszám-generátort.

Tesztelés

A modell ellenőrzése egységtesztekkel valósult meg.

MinefieldUnitTests tesztosztály tesztmetódusai:

- *MinefieldModelStartNewGameTest*: Új játék beállításai.
- *MinefieldTestPausing*: Szüneteltetés és szüneteltetés feloldása.
- *MinefieldModelMovementTest*: Játékos karakter mozgatása.
- *MinefieldPausedMovement*: A mozgás fel van függesztve amíg a játék szünetel.
- *MinefieldMovementBoundsTest*: A mozgás a játéktérre van korlátozva.
- *MinefieldTimerTickTimeAdvanceTest*: Az időzítő tick eseményét kezeli a modell, telik a játékidő.
- *MinefieldMinesDescendingTest*: Az idő telésére helyesen esnek az aknák.
- *MinefieldGameOverMovementTest*: Játék vége bekövetkezik és a megfelelő esemény kiváltódik, ha a játékos „nekimegy” egy aknának.
- *MinefieldGameOverMineDescendingOnPlayerTest*: Játék vége bekövetkezik és a megfelelő esemény kiváltódik, ha a játékos karakterre „ráesik” egy akna.
- *MinefieldMineSpawningIncreasingTest*: Új akna megjelenésének az esélye nő ahogy telik a játékidő. A teszthez mock használatával szimuláljuk a véletlenszám-generátort, hogy mindig a felső és alsó korlát számtani közepét adja vissza.
- *MinefieldSaveGameCalledTest*: A perzisztenciában definiált mentés végrehajtódik.
- *MinefieldLoadGameCalledTest*: A perzisztenciában definiált betöltés végrehajtódik.
- *MinefieldSaveGameFailedTest*: A mentés közbeni hiba kiváltóik.
- *MinefieldLoadGameFailedTest*: A betöltés közbeni hiba kiváltóik.
- *MinefieldLoadGameTest*: Adott adatok helyesen betöltődnek a modellbe.

A mentést és betöltést mock használatával szimuláljuk.

A MinefieldUnitTests osztály implementálja az IDisposable iterfészt, mert a MinefieldGameModel típusú _testModel field-je Disposable.