

# Eseményvezérelt Alkalmazások Fejlesztése

3. beadandó/15. feladat

Gonda Dávid

BIXU0S

gonda.david18@gmail.com

## Feladat

### Aknakereső

Készítsünk programot, amellyel az aknakereső játék két személyes változatát játszhatjuk.

Adott egy  $n \times n$  mezőből álló tábla, amelyen rejtett aknákat helyezünk el. A többi mező szintén elrejtve tárolják, hogy a velük szomszédos 8 mezőn hány akna helyezkedik el.

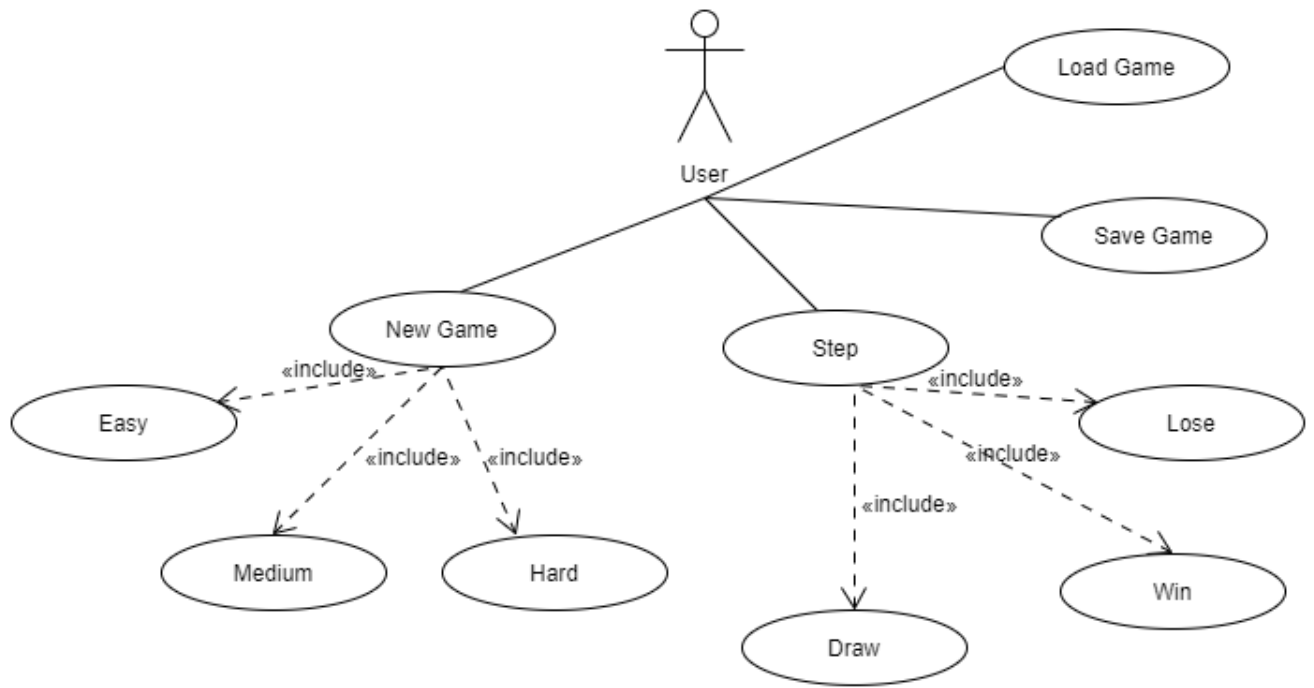
A játékosok felváltva léphetnek. Egy mező felfedjük annak tartalmát. Ha az akna, a játékos veszített. Amennyiben a mező nullát rejt, akkor a vele szomszédos mezők is automatikusan felfedésre kerülnek (és ha a szomszédos is nulla, akkor annak a szomszédai is, és így tovább). A játék addig tart, amíg valamelyik játékos aknára nem lép, vagy fel nem fedték az összes nem akna mezőt (ekkor döntetlen lesz a játék).

A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával ( $6 \times 6$ ,  $10 \times 10$ ,  $16 \times 16$ ), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött (ha nem döntetlen).

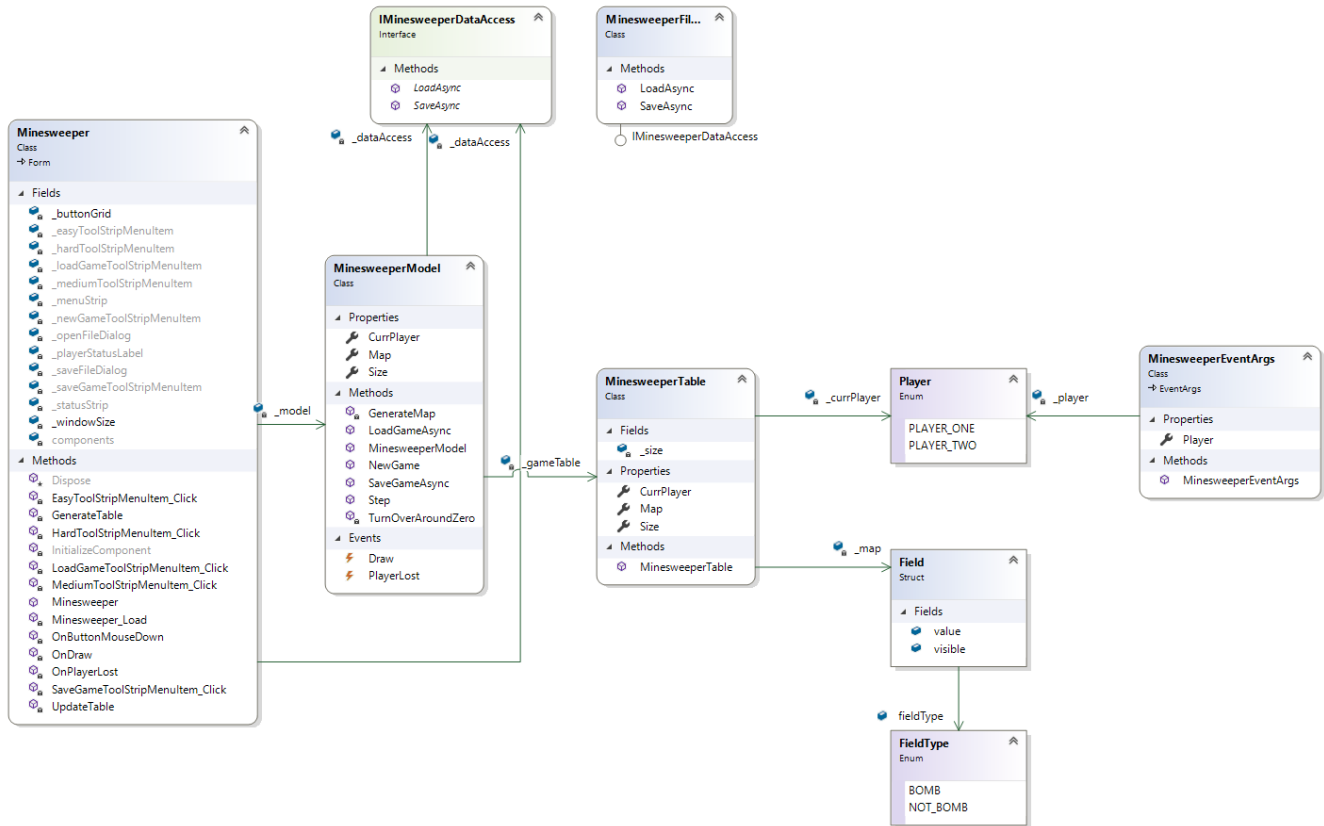
## Elemzés

- A játékot 3 nehézségi szinten lehet játszani, könnyűn ( $6 \times 6$ -os pálya), közepesen ( $10 \times 10$ -es pálya), és nehezen ( $16 \times 16$ -os pálya).
- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: New Game (Easy, Medium, Hard), Save Game, Load Game. Az ablak alján egy státuszsorba kiírjuk az aktuális játékost.
- A játéktáblát dinamikusan létrehozott nyomógombokkal fogjuk megvalósítani. Ezek alapból fehér gombok, kattintás után felfedjük mi van alatta.
- Bomba esetén az aktuális játékos veszít, ez után új játék kezdődik az aktuális nehézséggel.
- Nulla esetén a környező mezőket is felfedjük (ha ott is van nulla, ugyanígy járunk el).
- Abban az esetben, ha már csak bomba van lefedve a táblán, a játék döntetlennel végződik, ez után új játék kezdődik az aktuális nehézséggel.

- Jobb egérgommbal egy mezőt megjelölhetünk, ezzel jelezve a bomba veszélyét.



# Osztálydiagram



## Tesztelés

- A modell funkcionalitását egységtesztek segítségével ellenőriztük a **MinesweeperTest** osztályban. Ehhez létrehoztunk egy mock perzisztenciát (egy mock táblával együtt).
- Az alábbi tesztesetek kerültek megvalósításra:
  - o **MinesweeperModelNewGameTest:**  
Új játék indításának az ellenőrzése.
  - o **MinesweeperModelStepTest:**  
A lépésnek, annak hatásainak ellenőrzése.
  - o **MinesweeperModelTurnOverAroundZeroTest:**  
Nullára való lépés után a mező körül lévő mezők felfedésének ellenőrzése, több egymás melletti nullával.
  - o **MinesweeperModelGenerateMapTest:**  
Tábla generálásának az ellenőrzése.