

## Készítette:

Ömböli

Csaba

E-mail cím: [omblics@gmail.com](mailto:omblics@gmail.com)

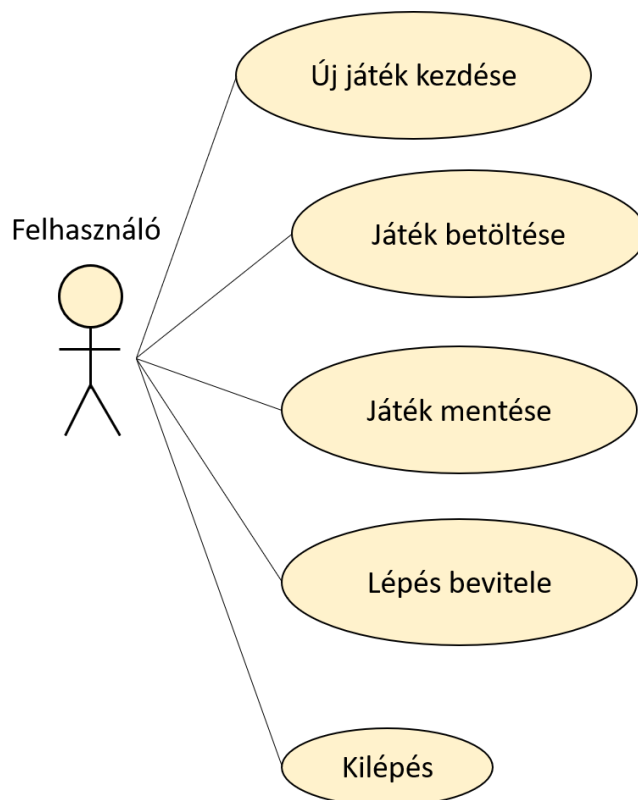
## Feladat

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy  $n \times n$  mezőből álló tábla, amelyen két játékos úrhajói helyezkednek el, középen pedig egy fekete lyuk. A játékos  $n - 1$  úrhajóval rendelkezik, amelyek átlóban helyezkednek el a táblán (az azonos színűek egymás mellett, ugyanazon az oldalon). A játékosok felváltva léphetnek. Az úrhajók vízszintesen, illetve függőlegesen mozoghatnak a táblán, de a fekete lyuk megzavarja a navigációjukat, így nem egy mezőt lépnek, hanem egészen addig haladnak a megadott irányba, amíg a tábla szélé, a fekete lyuk, vagy egy másik, előtte lévő úrhajó meg nem állítja őket (tehát másik úrhajót átlépni nem lehet). Az a játékos győz, akinek sikerül úrhajóinak felét eljuttatnia a fekete lyukba.

A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával ( $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ), valamint játék mentésére és betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött.

## A feladat elemzése

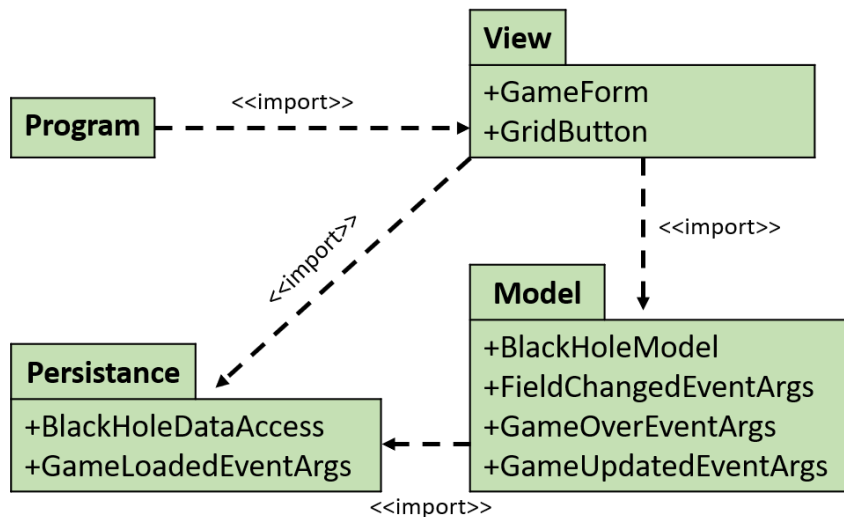
A játékban 2 játékos van, de a program szempontjából egy felhasználó létezik. A felhasználó egérrel kezelheti a programot.



## Tervezés

Programszerkezet:

- A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a View, a modell a Model, míg a perzisztencia a Persistence névtérben helyezkedik el. A névterek kapcsolatát a következő ábra írja le:



Perzisztencia

- A Persistence osztály biztosítja az adatok kezelését. A segítségével képesek vagyunk elmenteni a megkezdett játékot (`saveGame()`), vagy betölteni egy már korábban elmentettet.
- A `saveGame` a modelltől fogadja az elmentendő adatokat (a játéktáblát, a játékosokállását, és az éppen soron lévő játékost), majd egy szövegfájlba elmenti a játék adatai. A szövegfájl tartalmát (ha volt) felülírja.
- A `loadGame` metódus annak a szövegfájlnek az elérési útját fogadja paraméterként, ahonnan megpróbálja betölteni a játékot. Ha sikeres a betöltés, akkor a `GameLoaded` eseményen keresztül átadja a modellnek a betöltött adatokat, és igaz értékkel terminál. Ha a betöltés során hiba lép fel, a metódus hamis értéket ad vissza.

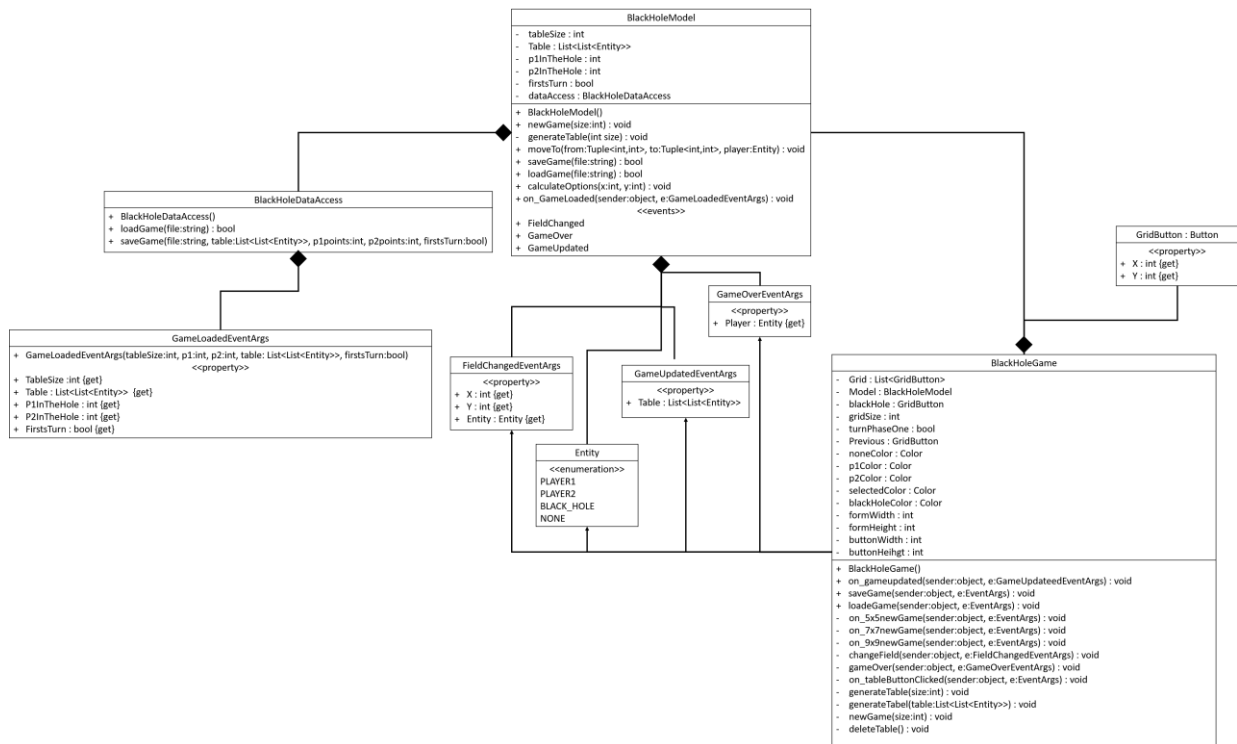
Modell

- A model egy mátrixban tárolja az aktuális játéktáblát (`System.Collections.Generic.List` tárolók egymásbaágyazásával) ez a `table`. A mátrix egyes elemei az itt bevezetett Entity enumeráció tagjai.
- Egy Entity típusú objektum az alábbi értékeket veheti fel: `PLAYER1`, `PLAYER2`, `BLACK_HOLE`, `NONE`
- A `saveGame` és a `loadGame` publikus metódusok meghívják az aggregált adatelérési réteg hasonló nevű metódusait.
- A `generateTable` tölti fel megfelelő értékekkel a táblát
- A `calculateOptions` `x` és `y` koordináták alapján a `fieldChanged` eseményen keresztül jelzi a nézetnek, az adott helyről lehetséges lépési opciókat.
- A `moveTo` kicseréli a `from` koordinátán tartózkodó játékos pozícióját a `to` koordinátára

## Nézet

- A nézet aggregálja a modellt, és az eseményeinek megfelelő eseménykezelőket használ
- A táblát nyomógombokkal (GridButton) reprezentáljuk, melyeket egy FlowLayoutPanel-re helyezünk el. (A tábla négyzet alakjának megőrzéséhez az ablak méreteit fix mérettel limitáljuk.)
- A GridButton típus a System.Windows.Forms.Button osztályból származik, és kiegészül x, és y koordinátákkal

## Osztálydiagram



sender	event	receiver	event handler
model	FieldChanged	nézet	changeField
model	GameUpdated	nézet	on_GameUpdated
model	GameOver	nézet	gameOver
dataAccess	GameLoaded	model	on_GamLoaded

## Tesztelés

A modell funkcionális egységesztek segítségével lett ellenőrizve a UnitTest1 osztályban

A tesztesetek listája:

- newGame5 : Új játék létrehozása 5x5-ös táblával
- newGame7 : Új játék létrehozása 7x7-ös táblával
- newGame9 : Új játék létrehozása 9x9-ös táblával

- saveGame: Játék mentése fájlba
- loadGame: Játék betöltése fájlból
- moveTest1: Lépések tesztelése
- moveTest2: Lépések tesztelése
- moveTest3: Lépések tesztelése