

3. Beadandó feladat dokumentáció

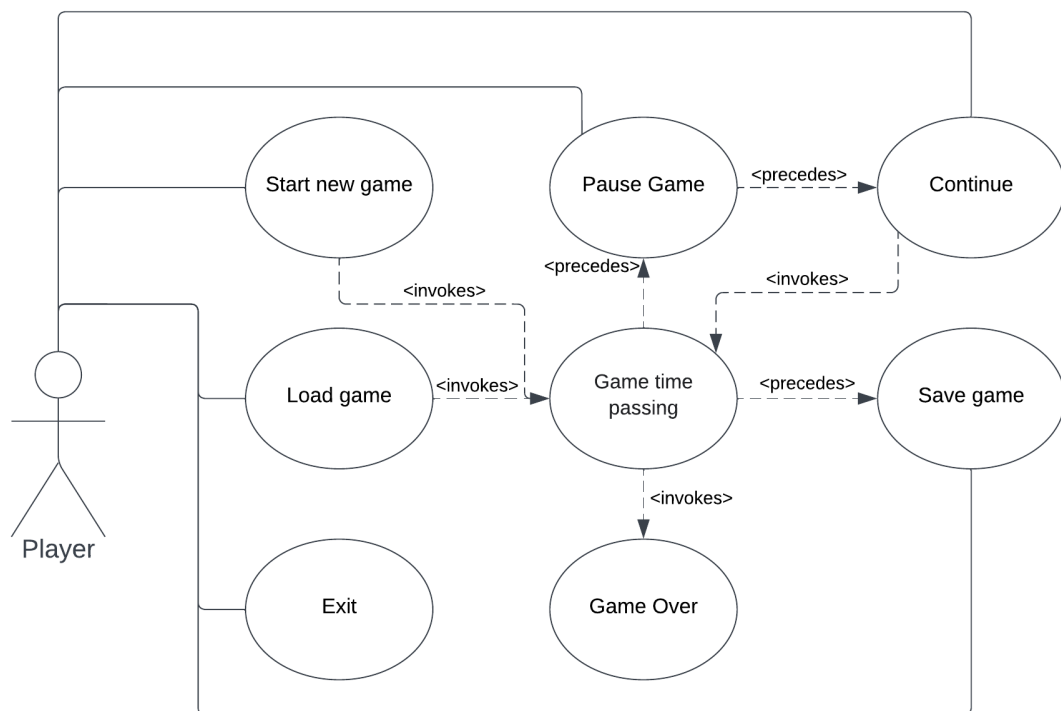
Készítette: Kertész János

E-mail: am2vz8@inf.elte.hu

Feladat:

Készítsünk programot, amellyel az aszteroidák játékot játszhatjuk. A feladatunk az, hogy egy űrhajó segítségével átnavigáljuk egy aszteroidamezőn. Az űrhajóval a képernyő alsó sorában tudunk balra, illetve jobbra navigálni. A képernyő felső sorában meghatározott időközönként véletlenszerű pozícióban jelennek meg az aszteroidák, amelyek folyamatosan közelednek állandó sebességgel a képernyő alja felé. Az idő múlásával egyre több aszteroida jelenik meg egyszerre, így idővel elkerülhetlenné válik az ütközés. A játék célja az, hogy az űrhajó minél tovább elkerülje az ütközést. 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.

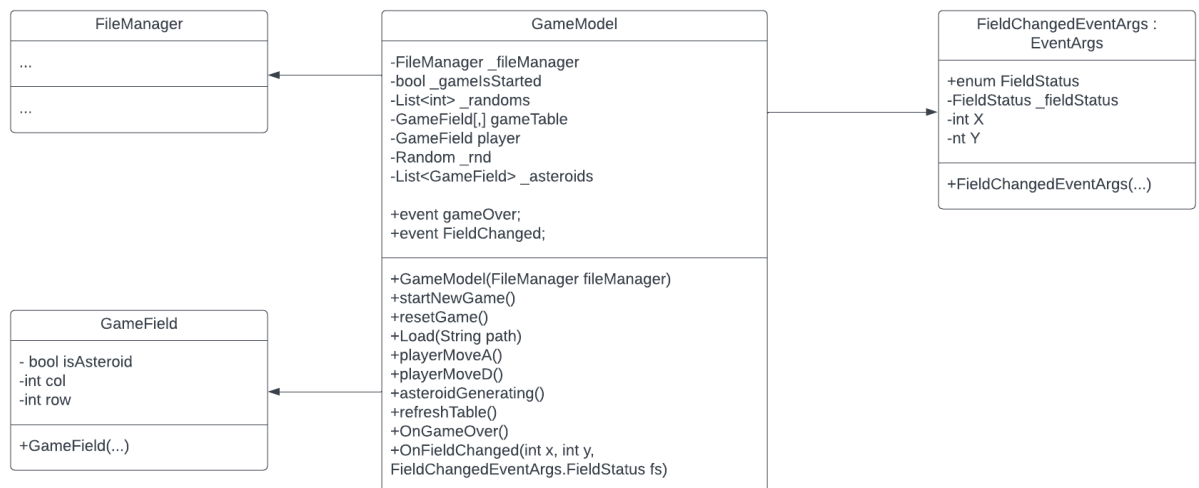
Elemzés:



Felhasználói esetek diagramja

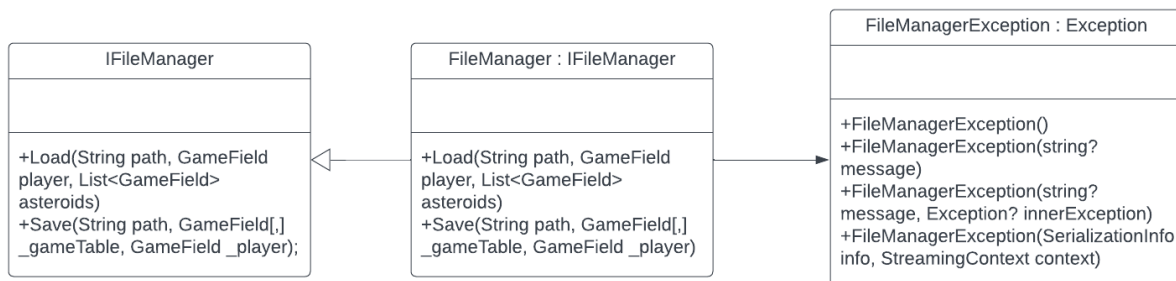
Tervezés:

- **Programszerkezet:**
 - A szoftvert két projektből építjük fel: a modellt és a perzisztenciát tartalmazó osztálykönyvtárból (.NET Standard Class Library), valamint a .NET MAUI többplatformos projektből, amelyet Windows és Android operációs rendszerre is le tudunk fordítani
 - A programot MVVM architektúrában valósítjuk meg, ennek megfelelően View, Model, ViewModel névttereket valósítunk meg az alkalmazáson belül. A program környezetét az alkalmazás osztály (AppShell) végzi, amely példányosítja a modellt, a nézetmodellt és a nézetet az App-on keresztül, biztosítja a kommunikációt, valamint felügyeli az adatkezelést. A program szerkezetét két projektre osztjuk implementációs megfontolásból: a **Persistence** és **Model** csomagok a program felületfüggetlen projektjében, míg a **View** egy másik csomagban kap helyet.
- **Model:**
 - A modell lényegi részét a GameModel osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, pl a hajó irányítását (playerMoveA, playerMoveD). A típus lehetőséget ad új játék kezdésére (startNewGame).
 - A pályán lévő meteorok generálását, és léptetését a (asteroidGenerating) és (refreshTable) függvények végzik. Mind a kettő tartalmaz egy elágazást, melyben ha a meteor a játékoshoz és, a (_gameOver) mező igazra vált, ezzel a játék megszakítását, és égét kiváltva.



- Perzisztencia:

- Az adatkezelés feladata a táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása
- A hosszú távú adattárolás lehetőségeit az FileManager osztály adja meg, amely lehetőséget ad a tábla betöltésére (Load), valamint mentésére (Save).
- A FileManager osztály saját (FileManagerException : Exception) kivétel osztállyal rendelkezik.
- A program az adatokat szöveges fájlként tudja eltárolni, melyek az .txt kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.
- A fájl 11 db 11 hosszú stringet tárol (ezek '\n'-el vannak elválasztva), melyekben '0'-val az üres, '1'-el a meteor '2'-vel a játékosra hivatkozó mezőket tároljuk.

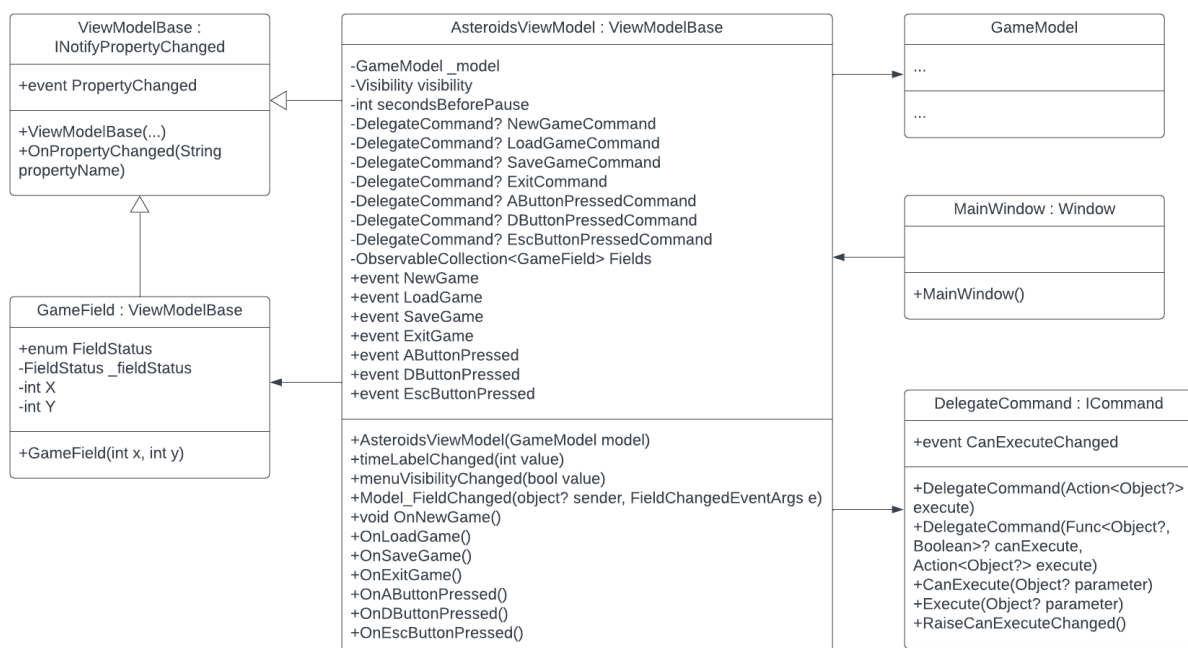


- View:

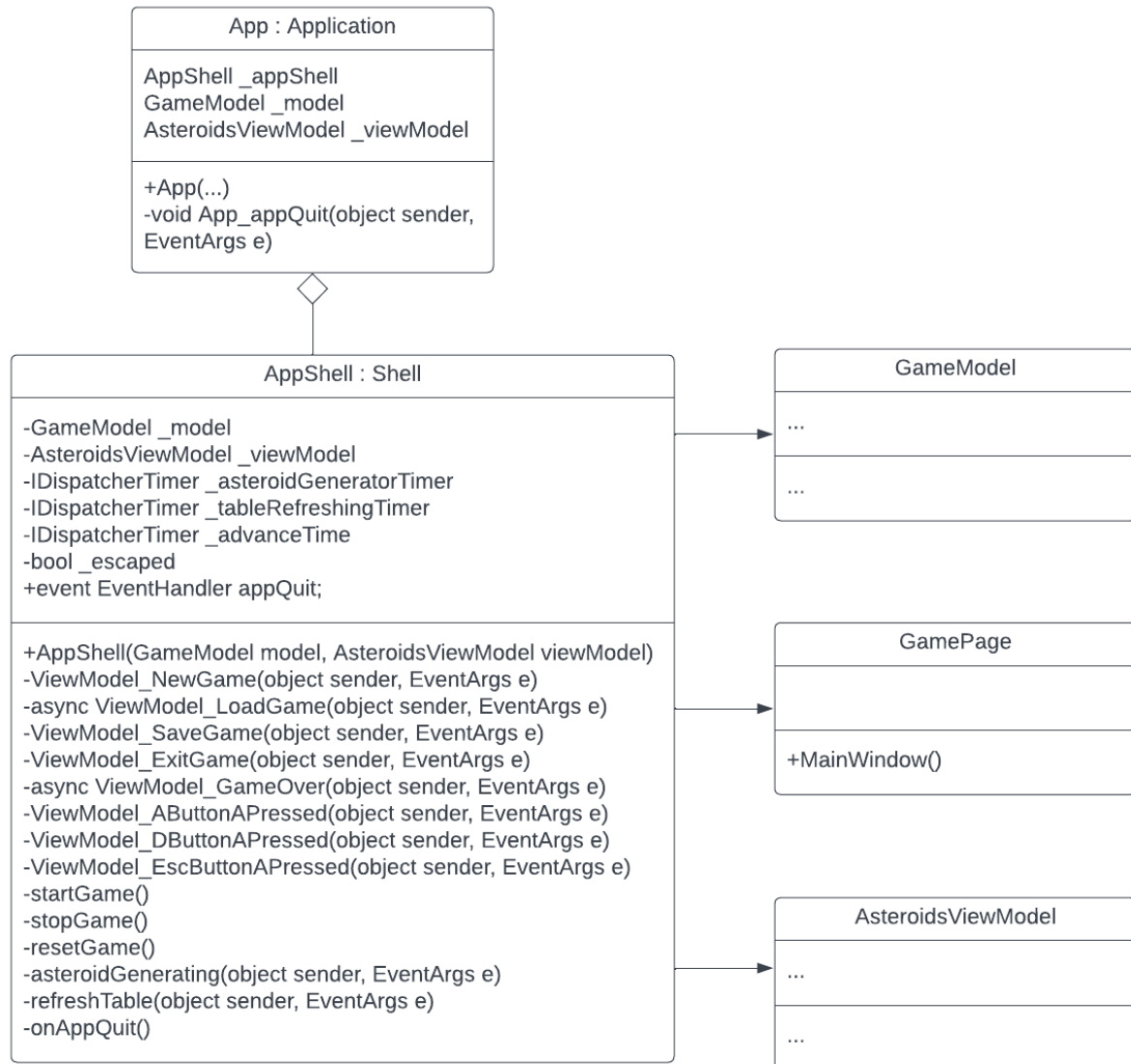
- A view réteg feladata a játék logikájának megjelenítése, valamint annak eseményekkel való összekötése.
- A nézetet a Form1 osztály biztosítja, amely tárolja a modell egy példányát (_gameModel), valamint az adatelérés példányát (_fileManager).
- A játéktáblát egy gombmező (_buttonGrid) reprezentálja. A felületen létrehozuk a megfelelő menüpontokat, illetve státuszsort, valamint dialógusablakokat, és a hozzájuk tartozó eseménykezelőket. A játéktábla generálását (tableSetUp) végzi
- A játék időbeli kezelését két időzítő végzi (_asteroidGeneratorTimer, _tableRefreshingTimer), amelyeket mindig aktiválunk játék során, illetve inaktíválunk, amennyiben bizonyos menüfunkciók futnak..

- ViewModel:

- A nézetmodell megvalósításához felhasználunk egy általános utasítás (DelegateCommand), valamint egy ős változásjelző (ViewModelBase) osztályt.
- A nézetmodell feladatait a SudokuViewModel osztály látja el, amely parancsokat biztosít az új játék kezdéséhez, játék betöltéséhez, mentéséhez, valamint a kilépéshez. A parancsokhoz eseményeket kötünk, amelyek a parancs lefutását jelzik a vezérlőnek. A nézetmodell tárolja a modell egy hivatkozását (_model), de csupán információkat kér le tőle. Direkt nem avatkozik a játék futtatásába.
- A játékmező számára egy külön mezőt biztosítunk (GameField), amely eltárolja a pozíciót, szöveget, engedélyezettséget. A mezőket egy felügyelt gyűjteménybe helyezzük a nézetmodellbe (Fields).



- App/AppShell:
 - Az App/ AppShell osztály feladata az egyes rétegek példányosítása (App_Startup), összekötése, a nézetmodell, valamint a modell eseményeinek lekezelése, és ezáltal a játék, az adatkezelés, valamint a nézetek szabályozása.
 - A játék léptetéséhez tárol egy időzítőt is (_advanceTime), amelynek állítását is szabályozza az egyes funkciók hatására.



Tesztelés:

A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a **GameLogicsTests** osztályban.

Az alábbi tesztesetek kerültek megvalósításra:

- **startNewGameTest**
Ellenőrzi a tábla méretének, és a játékos pozíciójának helyes beállítását.
- **asteroidGenerating_resetGameTest**
Ellenőrzi az aszteroidák generálódását. Valamint a tábla visszaállítása során az aszteroidák helyes törlését, valamint a játékos kezdő pozícióba állítását.
- **playerMoveA_playerMoveDTest**
Ellenőrzi a játékos mozgásának helyességét.
- **refreshTableTest**
Ellenőrzi az aszteroidák hullását, és azok eltűnését amint a pálya aljára értek.
- **GameOverFromAboveTest**
Ellenőrzi, hogy amint a játékost felülről aszteroida találja el a játék végét jelző boolean true lesz-e.
- **GameOverFromNextToTest**
Ellenőrzi, hogy amint a játékost oldalról aszteroidába ütközik, a játék végét jelző boolean true lesz-e.