

2.Beadandó feladat dokumentáció

Készítette:

Nagy Levente (uji470)

E-mail: levente0517@gmail.com

Feladat:

8. Aszteroidák

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 moulásával egyre több aszteroida jelenik meg egyszerre, így idővel elkerülhetetlenné válik az ütközés. A játék célja az, hogy az űrhajó minél tovább elkerülje az ütközést.

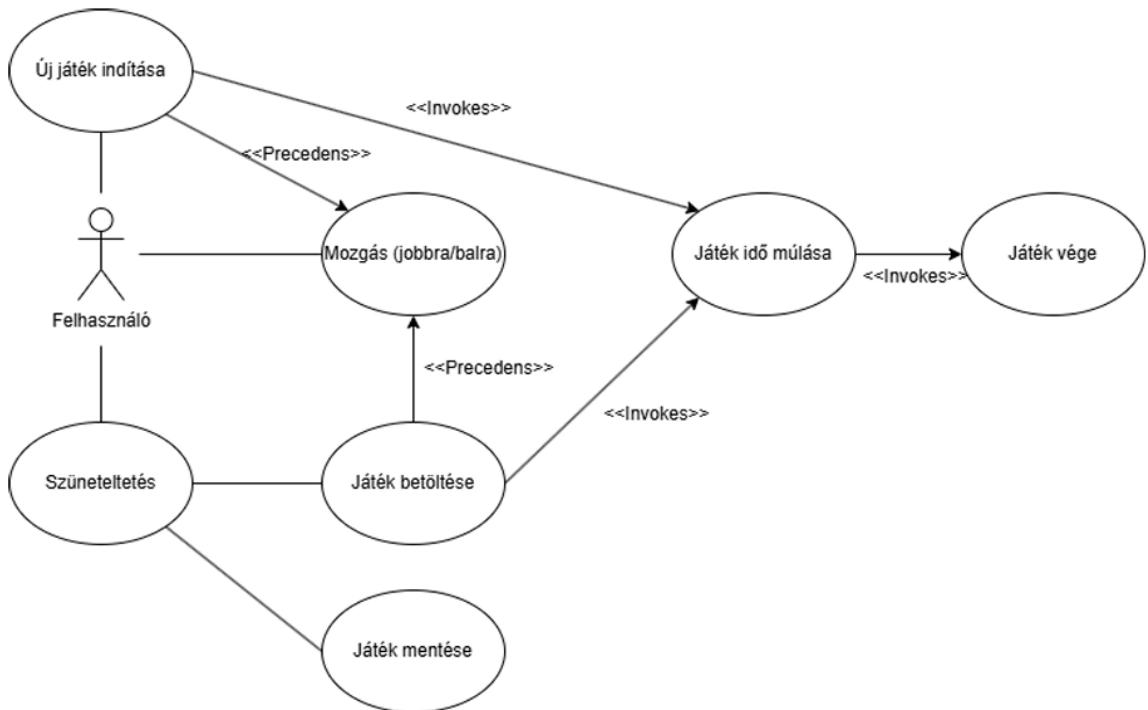
A program biztosítson lehetőséget új játék kezdésére, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog semmi a játékban). Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

Elemzés:

- A feladat megvalósítására **MVVM** (Model-View-ViewModel) architektúrát használunk és a View-t Microsoft Wpf application-ben valósítjuk meg.
- A játék elindításakor egy fix screen pop up-ol, amelyen szerepelnek az alábbi dolgok:
 - Felső panelban egy **File** gomb amely tartalmazza a következőket: **Save** (játék elmentése) és **Load** (játék betöltése fájlból).
 - File gomb mellett egy **New game** gomb, amellyel új játékot kezdünk
 - Valamint egy **Pause** gomb mely megállítja a játékot és lehetővé teszi, hogy használjuk a **File** gomb alatt elérhető dolgokat.
 - A játéktér amelyben megjelenik a Hajónk legalul és a játék kezdetével majd aszteroidák a képernyő tetején.
 - Az alsó panelban pedig 2 számláló melyek az eltelt időt és az új aszteroidák megjelenésének esélyét mutatja

- A játék végekor pedig egy screen amely megmutatja hogy mennyi ideg éltünk túl, ilyenkor a játék leáll és csak a **New Game** gomb megnyomásával tudjuk folytatni.

Felhasználói esetek:



1. Ábra felhasználói esetek diagramja

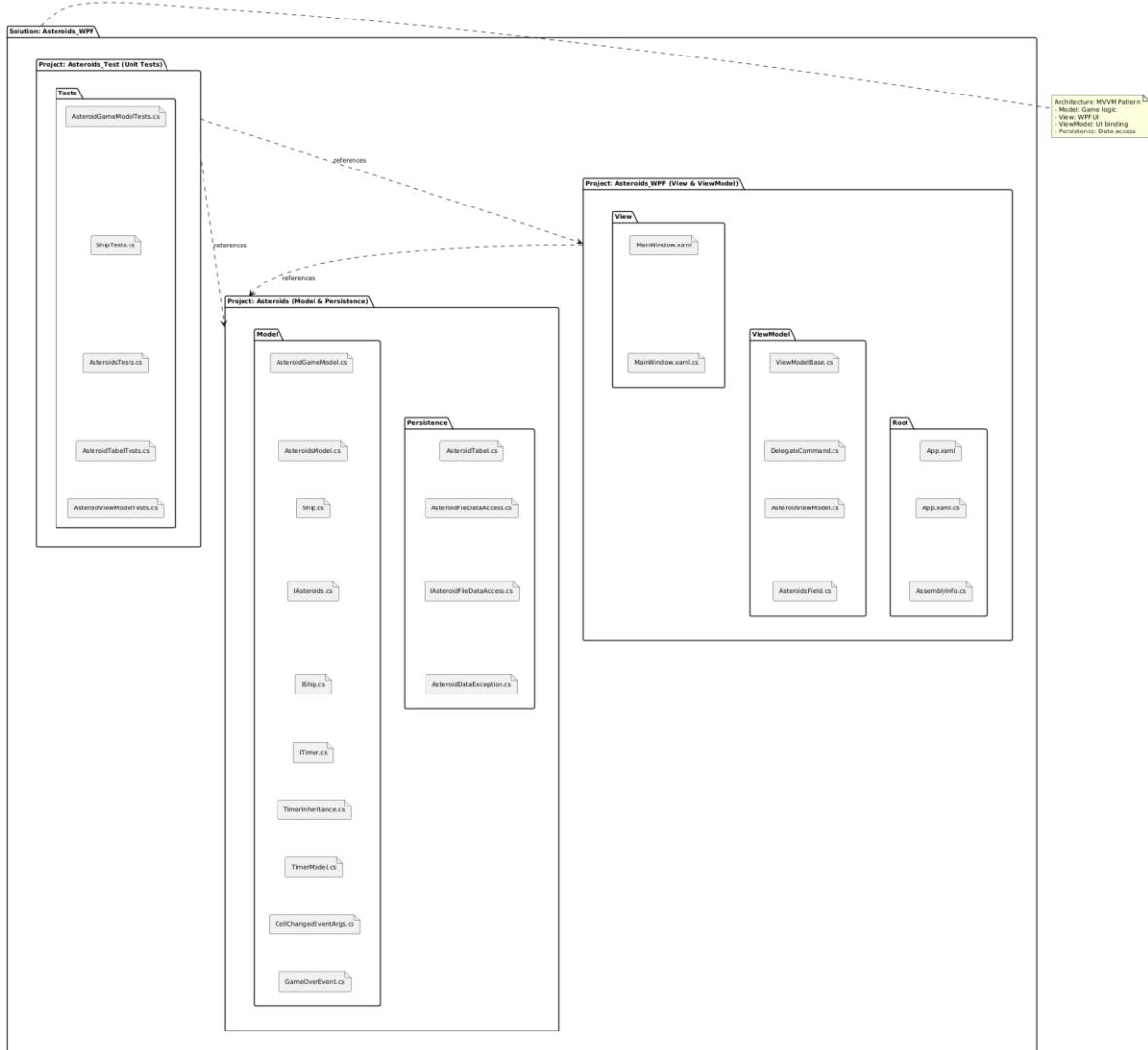
Tervezés:

- **Programszerkezet**
 - A játékot az említett MVVM architektúrában valósítjuk meg, azaz van egy **Model** (+ perzisztencia) részünk, egy **View** valamint egy **ViewModel**.
- **Model:**
 - A modelban az alábbi osztályok szerepelnek:
 - **AsteroidGameModel**: Feladata a játék logikájának kezelése
 - **AsteroidsModel** : Az aszteroidák osztálya
 - **IAsteroids**.: Az aszteroidák osztály interféce
 - **Ship**.: A hajó osztálya
 - **IShip**.: A hajó osztály interféce
 - **TimerModel**: A játék során használt időzítő osztálya

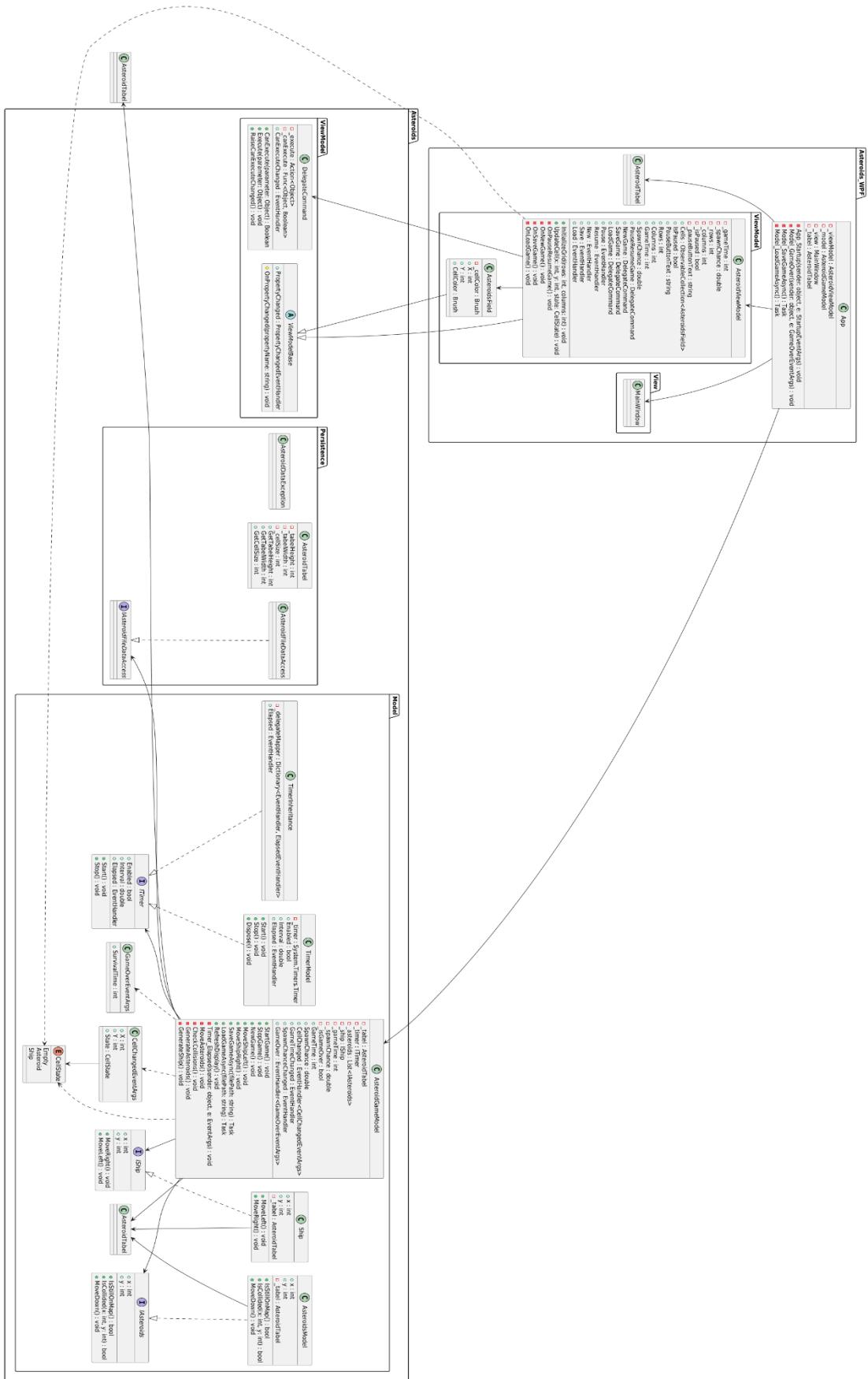
- **TimerInheritance:** A timer kezelése
- **ITimer:** A timer interféce
- **GameOverEvent:** A játék végét kiváltó event kezelése
- **CellChangedEventArgs:** A mezők változásának kezelője

- **Persistance:**
 - Az alábbi osztályokkal rendelkezik:
 - o **AsteroidDataException**: Az exception-ok kezelése
 - o **AsteroidFileDataAcces**: A játék filban eltráolását kezeli
 - o **AsteroidTabel**: A tábla felépítése
 - o **IAsteroidFileDataAcces**: A fájleltárolás interféce
- **View:**
 - Feladta a Window megvasítása itt szerepel a MainWindow.xaml és annak a .cs kódja
- **ViewModel**
 - Feladta a View és a Model összekapcsolása, itt az alábbi osztályok szerepelnek:
 - o **AsteroidsField**: A játékteret köti össze a modellel
 - o **AsteroidViewModel**: a Játék logikáját köti össze a modellel
 - o **DelegateCommand**
 - o **ViewModelBase**

Valamint Szerepel még a Az **app.xaml** és annak a .cs kódja amely az események kiváltásának kezelésére felelős.



2. Ábra: Az alkalmazás csomagdiagramja



3. ábra: Az alkalmazás osztálydiagramja

Tesztesetek:

- A teszteknél az alábbi osztályok szerepelnek:
 - o **AsteroidsGameModelTest**: A játék logikáját teszteli
 - o **AsteroidsTest**: Az aszteroidákat teszteli
 - o **AstroidsViewModelTest**: A view modelt teszteli
 - o **AsteroidTabelTest**: A játékmezőt teszteli